PhD Thesis

Railway traffic optimization

Universidad de Castilla-La Mancha
May 2014

Author
José Luis Espinosa Aranda

Advisor
Dr. Ricardo García Ródenas
To Noelia
All models are wrong, but some are useful

George Edward Pelham Box
Abstract

Currently railway systems are one of the most important means of transport used by passengers or for freight transportation in developed and developing countries. This activity has grown in importance and complexity in urban and inter-urban contexts because of the increasing population and the continuous change of demand patterns.

One of the problems that must be solved in a railway system is the scheduling of the trains. This problem is known as the timetable setting problem and can be tackled following two main approaches: i) an on-line approach, which is focused on restoring the schedule when a disruption occurs in the system in real-time using rescheduling techniques, and ii) an off-line approach focused on the developing of a timetable from scratch or from an updated timetable.

This thesis deals with the train scheduling problem, proposing two new approaches for the on-line and off-line scheduling of a railway system.

With respect to the on-line context an Intelligent Transportation System composed of a conflict detection and a conflict resolution module is proposed. The conflict detection module is defined as a discrete event simulation model which represents the behaviour of the trains considering the planned schedule. The conflict resolution module models the railway system by means of the alternative graph concept and implements the First Come First Served and Avoid Most Critical Completion algorithms of the literature and a novel approach to solving this problem based on demand. Computational experiments for the real case study of the RENFE Cercanías Madrid network have been performed in order to test this methodology.

In the off-line context, a mesoscopic model is proposed for High-Speed Rail systems, following a nominal approach and attempting to maximize the benefits of the company. This model represents the supply of the railway network as a discrete event simulation model, and the behaviour of passengers based on a random utility framework, which leads us to propose a novel constrained nested logit model, CMNL, which considers that a passenger will select a service depending on the timetable, price, travel time and seat availability. CMNL adopts the entropy-maximizing approach to include non-linear constraints for a decision-maker in its formulation. We propose the use of Reproducing Kernel Hilbert spaces to represent the utilities. This issue is essential to dynamic choice modelling such as departure time modelling. The calibration of CMNL is also discussed. Due to its high computational cost, the complete model is solved using various metaheuristic methods, proposing a hybridization approach which combines the advantages of the basic algorithms. Computational experiments for the real case study of the Madrid-Seville corridor have been developed to test this methodology.
Acknowledgements

All ways have a beginning and an end, and this way would not have end without the help of a lot of people. In this acknowledgements I want to express my gratitude to all the people and institutions which have supported me during all these years of hard work.

I am very grateful to my advisor and friend Ricardo García Ródenas for its implication and all the work and good moments we have spend together for more than seven years.

Thanks to my colleagues of MAT and SMILE groups (Use, Tere, Maria Luz, Doroteo, Alfredo, Victor Hugo, Clemen, Matheus, Christian, Abril, Bea) for all this time and chats in the café.

Thanks to MariMamen y Javier Ruiz Lázaro for their help in the implementation of the software tools used for the study of the models presented in this thesis.

I want to thank my friends MariMamen, DPZ, Dani, Neus and Xema for all the moments together around Ciudad Real and the world.

Thanks to Cele, Felipe, and all my teammates and friends of CB Almagro, we will see in courts.

I want to thank my parents Maria José and Luis, and my brothers Javier and Angel for stay by my side all this time.

And finally I want to express my infinite gratitude and love to Noelia, just for all She is and what I am when we are together.

This research has been financed by the Ministerio de Economia y Competitividad of Spain with the TRA-2011-27791-C03-03 research project. Thanks also to the Escuela Superior de Informática of Ciudad Real, the Mathematics and Information Technologies and Systems departments and the Universidad de Castilla-La Mancha for all the support provided.
## Contents

1 Introduction  
1.1 Railway planning problem ........................................ 5  
1.2 Railway (re-)scheduling problem ................................. 6  
1.2.1 Rescheduling problem (on-line approach) ...................... 7  
1.2.2 Train timetabling problem (off-line approach) .............. 10  
1.2.2.1 Robust TTP planning .................................. 11  
1.2.2.2 Nominal TTP planning .................................. 12  
1.3 Objectives and contributions of this thesis ..................... 13  
1.4 Published and submitted articles ............................... 16  
1.5 Thesis outline .................................................. 17

2 A discrete event-based simulation model for real-time traffic management in railways  23  
2.1 Introduction .................................................... 23  
2.2 Past research ................................................... 25  
2.3 Formulation of the railway model ................................ 27  
2.3.1 Railway network ........................................... 28  
2.3.2 Service network ........................................... 30  
2.3.3 Simulation algorithm ....................................... 32  
2.4 Computational experiments ..................................... 37  
2.4.1 Case study of Line C5 ..................................... 37  
2.4.2 Study of alternative operation plans ......................... 42  
2.5 Conclusions .................................................... 44

3 A demand-based weighted train delay approach for rescheduling railway networks in real time 49  
3.1 Introduction .................................................... 49  
3.2 Past research ................................................... 50  
3.3 Problem formulation ............................................ 52  
3.3.1 Alternative graph .......................................... 52  
3.3.2 A weighted train delay based on demand approach ......... 55  
3.4 Heuristic algorithms ............................................ 59  
3.4.1 Avoid Most Delayed Alternative Arc (AMDAA) ............... 59  
3.4.2 Illustrative example ....................................... 62  
3.5 Computational experiments ..................................... 64  
3.5.1 Experiment 1 ............................................... 66  
3.5.2 Experiment 2 ............................................... 68  
3.6 Conclusions .................................................... 69

4 A framework for derivative free algorithm hybridization  79  
4.1 Introduction .................................................... 79  
4.2 The conceptual framework of hybridization ..................... 81  
4.3 Instances of hybrid CG algorithms ............................. 83  
4.3.1 Instances for $A_r$ ........................................ 84  
4.3.2 Instances for $A_c$ ........................................ 86

VII
4.3.3 Instances of the framework .................................. 88
4.4 Computational results ........................................... 89
  4.4.1 Design of the experiments ................................. 90
  4.4.2 Results ................................................... 91
4.5 Conclusions ................................................... 92

5 Constrained logit model: Formulation and calibration 99
  5.1 Introduction ................................................. 99
  5.2 Formulation of the constrained logit model ................. 101
    5.2.1 Formulation of the multinomial logit model as a maxi-
          mum entropy optimization problem ..................... 102
    5.2.2 The disaggregated constrained multinomial logit model .... 103
    5.2.3 The aggregate constrained multinomial logit model (CMNL) 104
    5.2.4 The constrained hierarchical multinomial logit .......... 106
  5.3 Model behaviour and utility ................................ 108
    5.3.1 Utility formulation ................................... 109
  5.4 Calibration issues ......................................... 111
    5.4.1 Overspecification ..................................... 113
      5.4.1.1 First source of over-specification .................. 113
      5.4.1.2 Second source of over-specification ............... 114
  5.5 Numerical analysis ....................................... 115
    5.5.1 Application of the constrained hierarchical multinomial
          logit model for railway service modelling ............ 115
    5.5.2 Case study ............................................ 118
    5.5.3 Estimation methods .................................... 119
    5.5.4 Study of the best solution obtained .................... 121
  5.6 Conclusions ................................................. 122

6 High-Speed railway scheduling 127
  6.1 Introduction ................................................. 127
  6.2 A model for HSR scheduling ................................ 130
    6.2.1 Supply model .......................................... 130
      6.2.1.1 Dwell time and overtaking constraints ............ 132
      6.2.1.2 Rolling stock constraint .......................... 135
      6.2.1.3 The discrete event algorithm ....................... 135
    6.2.2 Demand model and utilities ............................ 137
    6.2.3 Objective function ................................... 137
  6.3 Metaheuristic algorithm ................................... 138
  6.4 Computational experiments ................................ 138
    6.4.1 Case study ............................................ 138
    6.4.2 Experiment I: qualitative analysis of solution ......... 140
    6.4.3 Experiment II: test of parallelization strategies .... 141
  6.5 Conclusions ................................................. 144

7 Conclusions and future research 147
  7.1 Conclusions ................................................. 147
  7.2 Suggestions for further research .......................... 148

A Test Functions 165
Contents

B Case studies data and links 171
  B.1 Chapter 2 ........................................... 171
  B.2 Chapter 3 ........................................... 171
  B.3 Chapters 5 and 6 ................................... 172

C MATLAB-GAMS parallelization 173
  C.1 Problems ........................................... 174
  C.2 Solution proposed ................................. 175
    C.2.1 GAMS code ...................................... 175
    C.2.2 MATLAB code ................................... 177
List of Figures

1.1 Number of travellers and number of travellers per kilometre interurban context, (INE (2010)) ........................................ 5
1.2 Taxonomy of Railway (Re-)Scheduling Problems .................. 8
1.3 On-line approach main problems ................................. 8

2.1 ITS architecture ................................................. 24
2.2 Railway Model .................................................. 28
2.3 Example of node-link model of a railway track .................... 29
2.4 Example of the state of the signals at a particular moment on the C5 Line of Renfe Cercanías Madrid in the stations of San José de Valderas and Alcorcón .......................... 30
2.5 Conservation of speed and instant when there is a change of block section ...................................................... 33
2.6 The SIMEIFER tool .............................................. 38
2.7 Line C5 of Renfe Cercanías Madrid ................................ 39
2.8 Modification to Laguna station .................................... 39
2.9 Time-Station diagram for first scenario .......................... 40
2.10 Graph of the running of the base example ......................... 41
2.11 Computational cost of ITS with time horizon 1 hour every 10 minutes .............................................................. 42
2.12 Example of crossover zones .................................... 43
2.13 Conflicts/hour for alternative operation plans .................... 44

3.1 An example of a physical railway network and its associated alternative graph, fixed arcs represent track segments occupied on top and travel time underneath ........................................ 55
3.2 Example of pair of alternative arcs ............................... 58
3.3 Illustrative example notation .................................... 63
3.4 AMDAA alternative arc selection depending on demand ...... 64
3.5 Case study rail map .............................................. 64
3.6 The SOFGA tool ............................................... 65
3.7 Number of trains delayed per algorithm .......................... 69
3.8 Computational cost comparison ................................... 73

4.1 Hybridization of Algorithms .................................. 83
4.2 PSO+NM scheme .............................................. 89
4.3 Objective Function vs Evaluations for SA, NM and SA+NM ... 91

5.1 Hierarchic nested logit model .................................. 106
5.2 Schematic representation of CMNL calibration procedure .... 113
5.3 Hierarchical MNL demand for railway service choice ........ 115
5.4 Madrid-Seville corridor ....................................... 118
5.5 Evolution of the SPSO algorithm for all cases .................. 122
5.6 Utility estimation with fixed departure time (8:30) .......... 123
5.7 Utility estimation with fixed travel time (63 min) .............. 123
6.1 HSR-TTP timetable model based on user behaviour . . . . . . . 130
6.2 Queue system represented as events . . . . . . . . . . . . . . . . 133
6.3 Discrete event simulation model flow chart . . . . . . . . . . . . 134
6.4 Dwell time constraint . . . . . . . . . . . . . . . . . . . . . . . . . 135
6.5 Time-station graph obtained by simulation . . . . . . . . . . . . 139
6.6 Performance of algorithms . . . . . . . . . . . . . . . . . . . . . . 141
6.7 Sequential SPSO . . . . . . . . . . . . . . . . . . . . . . . . . . . . 142
6.8 Parallel SPSO . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 143
6.9 Speed-up ratio for SPSO and SPSO+NM algorithms . . . . . . 143
## List of Tables

1.1 Problems depending on PTPP phases and planning horizon ... 7  
1.2 Summary of related studies on TS and DM grouped by characteristics considered ... 10  
1.3 Robust TTP articles ... 11  
1.4 Cyclic nominal TTP articles ... 13  
1.5 Non-Cyclic nominal TTP articles ... 14  

2.1 Results of the simulation for the standard timetable on Line C5 ... 40  
2.2 Results of simulation for different plans ... 44  

3.1 Relative error summary of results for 1-hour interval ... 67  
3.2 Relative error summary of results for 2-hour interval ... 67  
3.3 Numerical results for 1-hour interval (1/2) ... 70  
3.4 Numerical results for 1-hour interval (2/2) ... 71  
3.5 Numerical results for 2-hours interval ... 72  
3.6 Computational cost versus number of trains and amplitude of the time window ... 72  

4.1 Hybrid CG algorithm ... 84  
4.2 Nelder-Mead algorithm pseudocode ... 85  
4.3 SA algorithm ... 87  
4.4 SPSO algorithm ... 88  
4.5 NM, SA and SA+NM Results ... 93  
4.6 NM, PSO and PSO+NM Results ... 94  
4.7 Significances of Wilcoxon tests ... 95  

5.1 Examples of kernel functions ... 110  
5.2 Types of railway services on Madrid-Seville corridor ... 119  
5.3 Capacity of each type of train on Madrid-Seville corridor ... 119  
5.4 Solution found versus amplitude of feasible region ... 121  

6.1 Review of HSR disaggregate models (source: Cascetta and Coppola (2013)) ... 129  
6.2 Complete notation used ... 131  
6.3 Discrete event simulation algorithm ... 136  
6.4 Features of each service on Madrid-Seville corridor ... 138  
6.5 Potential total trips ($\hat{g}_w$) ... 139  
6.6 Improvements obtained by NM, SPSO and SPSO+NM algorithms ... 140  
6.7 Computational comparison between parallel algorithms ... 142
Railway Traffic Optimization
1. Introduction
1

Introduction

1.1 Railway planning problem

Railway transports are one of the most commonly used forms of public transports in developed countries, which includes regional transportation and long distance trips. These transport systems are having a very important impact on passenger mobility patterns due to the increasing number of trips and different types of services supplied.

Specifically, this form of transport has reached a state of high infrastructure development in Spain due to, among other factors, the "Plan Estratégico de Infraestructuras de Transporte" (PEIT) of the Development Ministry, which supports this means of transport with important funding until the year 2020. The three main aspects tackled in PEIT are high-speed rail, regional networks and freight transportation. According to the National Institute of Statistics of Spain, the Spanish railway networks transport approximately 700 million passengers per year, 645 million who use regional transportation in an urban context and 55 million who travel between cities in an interurban context. Figure 1.1 shows the number of travellers in interurban context by type of train.

![Figure 1.1: Number of travellers and number of travellers per kilometre interurban context, (INE (2010))](image-url)
In the same period 210 million passengers travelled by air, which shows the importance of railway transportation in this country. At the current moment (February 2014) the number of passengers of high-speed railways is greater than the numbers of passengers who uses air transportation.

Railway planning is a very complex process. Following the approach proposed by Ceder and Wilson (1986) any Public Transit Planning Problem (PTPP) can be split into four phases:

1. *Global line network planning.*
2. *Timetable setting.*
4. *Crew scheduling.*

Therefore, components of the railway system planning problem can be classified considering different planning horizons:

- *Strategical level* studies deal with the definition of the infrastructure of a railway system, the modification or expansion of this infrastructure and the rolling stock and crew management.

- *Tactical level* is focused on the modelling of the services which would use the railway network, considering line planning and timetabling problems. Furthermore at this level routing and allocation/assignment of the crew and vehicles are also tackled.

- The *operational level* manages and controls the daily state of the services in real time, considering rescheduling and re-routing solutions. Moreover the rolling stock circulation, rostering and shunting problems are dealt with at this level.

Table 1.1 summarizes these ideas and their relation to the PTPP phases. This PhD thesis is focused on solving the timetable setting problem at a tactical and operational level.

### 1.2 Railway (re-)scheduling problem

The growth of demand in railway transportation systems produce the necessity of increasing the capacity of the railway network, maintaining its quality and the level of service. Two possibilities exist to tackle this problem: i) to improve the railway infrastructures or ii) to design production based strategies.
1.2. Railway (re-)scheduling problem

<table>
<thead>
<tr>
<th>PTPP phases</th>
<th>Planning horizon</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global line network planning</td>
<td><strong>Strategical</strong></td>
</tr>
<tr>
<td>Timetable setting</td>
<td>Infrastructure definition</td>
</tr>
<tr>
<td>Vehicle scheduling</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Crew scheduling</td>
<td>Rollig stock management,</td>
</tr>
<tr>
<td></td>
<td>crew planning</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1.1: Problems depending on PTPP phases and planning horizon

The main problems of infrastructure modification are the economic cost of this approach and the geographical limitations. Thus, these limitations lead to the necessity of implementing production based strategies focused on the improvement of the available resources. The fast implementation and reduced investment are the main advantages of this approach compared with infrastructure modifications.

As a consequence of the previous comments analytical tools capable of aiding controllers in the decision making process are needed. Many studies are currently being carried out to solve railway optimization problems. We refer the reader to Caprara et al. (2007), Caprara et al. (2002) and Cordeau et al. (1998) for surveys in railway optimization.

The main methods followed to schedule railway systems are divided into on-line and off-line approaches. Real-time planning (on-line) is addressed through rescheduling. In this problem an initial timetable exists and the objective is to optimize the schedule if a conflict occurs. The off-line approach is focused on solving the Train Timetabling Problem (TTP), creating an optimal timetable from scratch or improving a previously defined schedule. Figure 1.2 shows roughly the taxonomy of this problem related to the uncertainty level of a railway system’s current state and the required response time. Later these two approaches will be explained in detail.

1.2.1 Rescheduling problem (on-line approach)

Railway systems are planned considering the possible perturbations that could occur with the objective of reducing the consequences of these disruptions in the schedule. Even so, it is practically impossible to generate a completely robust and reliable timetable capable of dealing with all types of disruption (see Vromans et al. (2006)).

In the case that a disruption occurs in the planned schedule it will
be necessary to solve the occasioned conflicts in the best possible way in
minutes, which is considered the on-line approach. In this context the
problems typically dealt with are conflict detection and the resolution of these
detected conflicts (see Figure 1.3).

Recently the conflict detection problem has attracted great attention
since several railway networks are using new technologies that allow the
dispatcher to know the position and speed of the trains in real time. This
information can be used to develop analytic tools capable of anticipating
conflicts of rail traffic which can cause delays in the railway system. One
approach to solving this problem in real time has focused on finding optimal
solutions to certain non-linear programming models Szpiegel (1973), Cordeau
et al. (1998). This type of methodology has two major disadvantages:

1. Despite the high computational capacity of modern computers, solving
a non-linear programming problem with constraints for each service
would involve a very high computational cost. For some problems this

Figure 1.2: Taxonomy of Railway (Re-)Scheduling Problems

Figure 1.3: On-line approach main problems
1.2. Railway (re-)scheduling problem

computational burden makes these methods unviable.

2. If a solution of this type is used, each time a train cannot fulfil its predetermined timetable, the entire plan must be recalculated from the current state of the network.

One solution recently proposed in the literature is based on simulation models (see Martin (1999), Ho et al. (2002), Baohua et al. (2007)), more exactly based on discrete events, incorporating into the models the concept of Travel Advanced Strategy (TAS) proposed by Dorfman and Medanic (2004). These models reduce the complexity of the calculations and eliminate the need to recalculate all the schedules each time a delay occurs. A similar approach is used in Li et al. (2008). These authors use a more effective strategy than the TAS, since they consider the speed of the train on a block section as a decision variable to represent different changes. Another approach is followed in Burdett and Kozan (2009) using compound buffers to maintain the correct occupancy levels of lines while allowing trains to pass through crossover points without additional routing decisions.

Currently new methodologies are being developed combining the described simulation models with optimization algorithms which try to aid controllers in their decisions (see Cacchiani et al. (2014)).

Bilevel programming is a general framework used for solving train-conflict resolution in real-time. The lower level problem defines an equilibrium between railway services’ supply and demand. The supply model represents the infrastructure, capacities, operating rules, safety rules and design of the train services of the rail network. This model provides the real timetable when a disruption occurs following a predetermined rescheduling strategy. The demand model represents user behaviour in the railway network.

The supply model is modelled as Train Scheduling (TS) at a microscopic level which represents how the trains move through the network. The demand model takes into account the decisions of passengers at a macroscopic level including re-routing or disconnecting strategies which consider the timetable and the possible connections between trains. This problem is known as Delay Management (DM).

The upper level problem represents rescheduling decisions within the railway network. The main points of view in the literature for modelling the objective function are the minimization of weighted train delays or the minimization of train delays. The first approach needs to include a demand model which increases its complexity when it is solved in real time. Because of this in most cases the approaches found in the literature are focused on the TS.

Table 1.2 shows a classification of the literature according to the approach followed.
### TRAIN SCHEDULING (TS) AND DELAY MANAGEMENT (DM)

<table>
<thead>
<tr>
<th>References</th>
<th>Train Scheduling (TS)</th>
<th>Delay Management (DM)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kraay and Harker (1999), Tornquist and Persson (2007), Min et al. (2011), D’Ariano et al. (2008a), Corman et al. (2010), Corman et al. (2011b)</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Mascis and Facciarelli (2002), D’Ariano et al. (2007a)</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Mazzarello and Ottaviani (2007)</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>D’Ariano et al. (2008b), Corman et al. (2011a), Corman et al. (2012b), Tornquist Krasemann (2012)</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>D’Ariano et al. (2007b), Corman et al. (2009)</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Dollevoet et al. (2009), Schachtebeck and Schöbel (2007)</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Corman et al. (2012a), Kanai et al. (2011), Dollevoet et al. (2012)</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Almodovar and Garcia-Rödenas (2013)</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Cadarso et al. (2013), Cadarso and Marin (2014)</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Wang et al. (2013)</td>
<td>X</td>
<td>Passenger comfort</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Fuel consumption</td>
</tr>
</tbody>
</table>

**Table 1.2:** Summary of related studies on TS and DM grouped by characteristics considered

#### 1.2.2 Train timetabling problem (off-line approach)

The large scale of the networks, limitations on resource sharing, interdependence between services and congestion makes off-line planning very complex. It is important to consider all these factors to develop a timetable without conflicts maintaining an equilibrium between the maximization of the capacity of the network and the robustness (see Landex et al. (2006)).

The TTP is a scheduling problem that tries to generate a timetable from scratch or considering a previous schedule, specifying the arrival and departure time of each train taking into account the imposed operating and safety constraints such as switches, track connections, station and platform characteristics, time windows, rolling stock, crew scheduling, train connections and platform allocation. The way to solve TTP could be addressed by two different approaches (see Cacchiani and Toth (2012)): robust planning that takes into account a set of scenarios in order to avoid future disruptions or nominal planning that solves the problem based on an objective function assuming a unique state of the railway system.
1.2. Railway (re-)scheduling problem

1.2.2.1 Robust TTP planning

The robustness of a railway system can be defined as the capacity to return the system to the planned schedule after disruptions occur. Robust TTP tries to take into account possible future delays at the operational level, trying to mitigate delay propagation when a disruption occurs. It could be addressed by including buffer times in the planning phase that insert empty periods of time between trains in timetables used to absorb a delay that might appear in the future.

In some cases the buffer times are not enough to return to the previous schedule, so it is necessary to introduce recovery times and waiting times in the stations during the programmed stops. Nie and Hansen (2005). Planned waiting time can lead to the increase of stopped or operational time. In some cases these introduced periods of time could cause the cancellation of a route, Wendler (2007).

The methods used in the literature for solving the TTP following the robust planning approach are stochastic programming, light robustness, recoverable robustness, delay management, bi-criteria Lagrangian approaches and meta-heuristics. Table 1.3 classifies the main existing literature on robust planning with respect to these six approaches.

<table>
<thead>
<tr>
<th>References</th>
<th>Approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kroon et al. (2008), Fischetti et al. (2009), Niu and Meng (2014)</td>
<td>Stochastic programming</td>
</tr>
<tr>
<td>Fischetti and Monaci (2009), Fischetti et al. (2009), Goerigk et al. (2013b)</td>
<td>Light robustness</td>
</tr>
<tr>
<td>Liebchen et al. (2009), Cicerone et al. (2009), D’Angelo et al. (2009), Cicerone et al. (2008b), Goerigk et al. (2013a)</td>
<td>Recoverable robustness</td>
</tr>
<tr>
<td>Liebchen et al. (2010), Cicerone et al. (2008a)</td>
<td>Delay management</td>
</tr>
<tr>
<td>Schöbel and Kratz (2009), Borndörfer and Schlechte (2008), Borndörfer and Schlechte (2007), Cacchiani et al. (2012)</td>
<td>Bi-criteria Lagrangian approach</td>
</tr>
<tr>
<td>Tormos et al. (2008a), Hanafi and Kozan (2014)</td>
<td>Meta-heuristics</td>
</tr>
</tbody>
</table>

Table 1.3: Robust TTP articles
1.2.2.2 Nominal TTP planning

This type of planning tries to calculate timetables for a given number of trains taking into account the constraints defined by the controllers, minimizing an objective function which represents the goodness of the solution. This method is focused on a specific scenario and it assumes that if there was a disruption, it should be solved in real-time.

This approach is usually associated with demand forecasting due to the interest of the controllers in knowing the behaviour of the passengers (Cascetta and Coppola (2013)).

Nominal TTP may be classified, see Cacchiani and Toth (2012), according to application: i) cyclic (periodic) or non-cyclic timetables, ii) railway network or single one-way line (corridor) linking two major stations with intermediate stations, iii) freight or passenger transportation and iv) objective function.

Cyclic timetabling was introduced by Serafini and Ukovich (1989), describing the so-called Periodic Event Scheduling Problem (PESP). Each event represents the arrival and departure from a given station. Another approach based on PESP is known as the Cycle Periodicity Formulation (CPF) based on the constraint graph proposed by Nachtigall (1994). Table 1.4 shows the current literature related to the type of model used and the method/approach presented in the paper for solving the model.

For the non-cyclic nominal planning problem, a first model using an Integer Linear Programming (ILP) formulation was proposed by Brännlund et al. (1998) which is solved using a Lagrangian relaxation approach.

Three main formulations of this problem are presented in the literature. The first is a mixed ILP formulation which represents the arrival and departure times by continuous variables, and the order of trains in each station is defined by binary variables (see Carey and Lockwood (1995)). The second approach is focused on the discretization of time and the representation of the problem on a time-space graph, where each node is the departure or arrival time of a train in a station, and the arcs represent the travel time between stations (see Caprara et al. (2002)). The last method proposes formulating the problem as a Constraint Satisfaction and Optimization Problem (CSOP). The variables are frequencies, arrival and departure times of trains at stations. The problem formulation is then transformed into a formal mathematical program to be solved for optimality by means of mixed integer programming techniques (see Barber et al. (2004) and Ingolotti et al. (2004)).

Another approach is the Timetable Planning and Rolling Stock Model (TTP&RSM) recently proposed by Cadarso and Marin (2012) focusing on the development of an integrated planning model to adapt the capacity offered and the system frequencies to attend to the increased passenger demand and...
1.3. Objectives and contributions of this thesis

The main objectives and contributions of the present research are:

O1. The study and implementation of the conflict detection and conflict resolution modules involved in an Intelligent Transportation System (ITS) for solving the rescheduling problem.

a) Conflict detection module. The objective is to develop a microscopic model capable of representing the system accurately with an acceptable computational cost. This model must consider the position of the trains at each instant and the safety constraints of the system (control signals, maximum speed tracks, etc.). This objective has led to the following

<table>
<thead>
<tr>
<th>Model</th>
<th>References</th>
<th>Method/approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>PESP</td>
<td>Schrijver and Steenbeek (1994)</td>
<td>Constraint propagation</td>
</tr>
<tr>
<td></td>
<td>Nachtigall and Voget (1996)</td>
<td>Passenger waiting time minimization</td>
</tr>
<tr>
<td>Extended PESP</td>
<td>Liebchen et al. (2007)</td>
<td>Heuristics (genetics, simulated annealing) and constraint programming procedures</td>
</tr>
<tr>
<td></td>
<td>Liebchen and Möhring (2007)</td>
<td>PESP integration with network and line planning</td>
</tr>
<tr>
<td></td>
<td>Kroon et al. (2013)</td>
<td>Flexible connections inclusion</td>
</tr>
<tr>
<td>CPF</td>
<td>Odijk (1996)</td>
<td>Cutting plane algorithm LP relaxation</td>
</tr>
<tr>
<td></td>
<td>Nachtigall (1998), Lindner (2000), Peeters</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(2003), Peeters and Kroon (2001), Lindner</td>
<td></td>
</tr>
<tr>
<td></td>
<td>and Zimmermann (2005)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Cordone and Redaelli (2011)</td>
<td></td>
</tr>
<tr>
<td>PESP, CPF</td>
<td>Kroon (2003)</td>
<td>Variable demand considered</td>
</tr>
<tr>
<td></td>
<td>and Peeters</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1.4: Cyclic nominal TTP articles

traffic congestion around urban and suburban areas.

Table 1.5 shows the current literature related to each type of model used and the method/approach presented in the paper for solving the model.

1.3 Objectives and contributions of this thesis
<table>
<thead>
<tr>
<th>Model</th>
<th>References</th>
<th>Method/approach</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Cai and Goh (1994)</td>
<td>Constructive heuristic</td>
</tr>
<tr>
<td></td>
<td>Higgins et al. (1997)</td>
<td>Various heuristics (local search, tabu search, genetic and hybrids)</td>
</tr>
<tr>
<td></td>
<td>Oliveira and Smith (2000)</td>
<td>Constraint programming</td>
</tr>
<tr>
<td></td>
<td>Narayanaswami and Rangaraj (2012)</td>
<td>Incorporate disruptions</td>
</tr>
<tr>
<td>Time-space graph of Caprara et al. (2002)</td>
<td>Caprara et al. (2006),</td>
<td>Heuristic Lagrangian relaxation</td>
</tr>
<tr>
<td></td>
<td>Cacciani et al. (2010)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Semet and Schoenauer (2005)</td>
<td>Evolutionary algorithm</td>
</tr>
<tr>
<td></td>
<td>Vansteenwegen and Oudheusden (2006)</td>
<td>Discrete events simulation</td>
</tr>
<tr>
<td></td>
<td>Cacciani et al. (2008)</td>
<td>LP relaxation using exact and heuristic methods</td>
</tr>
<tr>
<td>CSOP</td>
<td>Barber et al. (2004)</td>
<td>Pre-processes. Exact and heuristic methods</td>
</tr>
<tr>
<td></td>
<td>Ingolotti et al. (2004)</td>
<td>Sequential algorithm</td>
</tr>
<tr>
<td></td>
<td>Ingolotti et al. (2006)</td>
<td>Meta-heuristic algorithm</td>
</tr>
<tr>
<td></td>
<td>Salido et al. (2007), Abril et al. (2008)</td>
<td>Distributed approach</td>
</tr>
<tr>
<td></td>
<td>Tormos et al. (2008b)</td>
<td>Genetic algorithm</td>
</tr>
<tr>
<td></td>
<td>Salido and Barber (2009)</td>
<td>Constraint ordering heuristic</td>
</tr>
<tr>
<td>MILP (TTP&amp;RSM)</td>
<td>Cadarso and Marin (2012)</td>
<td>Exact branch and bound algorithm</td>
</tr>
</tbody>
</table>

Table 1.5: Non-Cyclic nominal TTP articles

Contributions:

- A clear description of the traffic control problem is given and a discrete event simulation model is proposed from the very beginning based on this description capable of detecting future conflicts in a railway system. The proposed simulation model is focused on urban rail systems such as regional or underground railways, but can also be used in inter-urban networks.

- Because of the low computational cost the simulation model described can be integrated as a part of an ITS as a schedule analyser or
1.3. Objectives and contributions of this thesis

for studying the feasibility of the control strategies such as sequencing trains, alternative routes or proper speed recommendations.

b) Conflict resolution module. The objective is to develop an on-line methodology for timetable rescheduling in case of disruptions or conflict detection capable of optimizing in real-time the disrupted schedule.

The main feature of this model is to introduce an estimation of the demand to the alternative-graph approach (D’Ariano (2008)) by means of a linear programming formulation which tries to minimize train delays, weighted by the number of passengers that arrive at each station. This approach is focused on eliminating the flaws of existing alternative graph models when a schedule is disrupted in various services at the same time, because it will only focus on the train with the biggest delay without taking into consideration the complete system. This objective function is a particular case of the non-linear cost function of delays at arrivals and departures presented by Kraay and Harker (1995). The model proposed tries to take into account implicitly the DM without treating demand lost at connections explicitly. Moreover the classical algorithm used for solving the alternative graph model is adapted to this new model.

Another contribution is that computational experiments have been carried out on the real local Renfe network in Madrid for both models, which proves the suitability for on-line use and thus for their incorporation into an ITS.

O2. The study and development of a methodology for solving the railway timetable problem based on random utility theory. The objective is to generate the timetable of the system following a nominal approach and maximizing the benefits of the company. This problem has been studied widely in the literature, but currently relatively limited attention has been given to High-Speed Railways (HSR) in context. The proposed model is divided into two sub-models:

a) A model which represents the services supplied for the railway network. The contribution of this work is the definition of a mesoscopic discrete event simulation model.

b) A demand model which represents and predicts the behaviour of the passengers. This thesis contributes to the state of the art proposing a random utility framework based on a novel constrained nested logit model which considers that a passenger will select a service depending on the timetable, price, travel time and seat availability. This model adopts the entropy-maximizing approach to bring the non-linear constraints for a decision-maker into its formulation. The main contribution, compared to the approaches described in the literature (see Anas (1983), Donoso and de Grange (2010), De Grange et al. (2013)), is that general utilities are considered, not necessarily with linear attributes. We propose the use of Reproducing Kernel Hilbert spaces to represent the utilities. This issue is essential to dynamic choice modelling such as departure time modelling.
The proposed model must be calibrated, so another contribution of this thesis is a new method for calibration based on the novel point of view of considering a subset of utilities as parameters for the estimation instead of classical weights of the attributes. This approach obtains the multi-attribute utility functions and they are used to calculate all the non-estimated utilities. All these facts produce the appearance of the following computational challenge:

O3. The study and development of algorithms capable of optimizing simulation-based functions. The models proposed in objective 2 have a high computational cost which can be tackled using free derivative optimization methods. This fact allows functions to be used which are implicitly defined by means of an optimization or simulation model. This thesis contributes to the state of the art proposing an extension of the column generation framework (see Garcia et al. (2003)) to derivative free optimization methods for general optimization problems.

1.4 Published and submitted articles

While this research work was being carried out, some related papers have been published or are at the review stage. They include papers read at conferences, articles in ISI journals and an award.

Related to objective 1.a, a preliminary study was published in the book Models and Technologies for Intelligent Transportation Systems associated to the International Conference on Models and Technologies for Intelligent Transportation Systems in 2009 (Espinosa-Aranda and García-Ródenas (2009)) and presented also in MORE 2009 conference (Espinosa-Aranda (2009)). An improved version was awarded by the Spanish Education Ministry with an Accesit in the VIII Certamen Universitario Arquímedes de Introducción a la Investigación Científica. Later, this improved work was presented in the MORE 2010 conference (Espinosa-Aranda (2010)) and finally published in the Journal of Intelligent Transportation Systems (Espinosa-Aranda and García-Ródenas (2012)).

Objective 1.b led to the publication of an article in the Journal of Rail Transport Planning and Management (Espinosa-Aranda and García-Ródenas (2013)).

The contributions of Objective 2 were presented in MOLC2013 (Espinosa-Aranda et al. (2013a)) and ELAVIO2013 (Espinosa-Aranda and García-Ródenas (2013)) conferences. An improvement of the work related to Objective 2.b has been submitted to the Transportation Research-B journal (Espinosa-Aranda et al. (2014b)) and an extension of the complete Objective 2 will be submitted to the European Journal of Operational Research (Espinosa-Aranda et al. (2014c)).
The results obtained from Objective 3 were presented to ICANNGA’13 conference and published in a *Lecture Notes in Computer Science* (Espinosa-Aranda et al. (2013b)). This approach is being used in various studies, included the two papers submitted related to objective 2 (Espinosa-Aranda et al. (2014b) and Espinosa-Aranda et al. (2014c)).

Furthermore the author of this document has participated in other research works and papers not directly related to this thesis. Among them it is worth emphasizing a paper focused on developing a modelling framework for the estimation of optimal CO2 emission taxes for private transport and published in *Procedia Social and Behavioral Sciences* (Almodóvar et al. (2011)), an article which proposes a Monte Carlo approach to simulate the stochastic demand in a continuous dynamic traffic network loading problem published in the *IEEE Transactions on Intelligent Transportation Systems* (Sánchez-Rico et al. (2014)) and a paper which is currently in its second revision in the *Annals of Operational Research* journal centred on presenting a Lagrangian relaxation approach for the expansion of a highway network (Angulo et al. (2014)).

### 1.5 Thesis outline

This work is divided into two main parts. *Part I* is focused in solving the rescheduling problem (on-line approach) and *Part II* is focused on solving the TTP (off-line approach). Each part is organized as follows.

**Part I** is composed of Chapters 2 and 3. Chapter 2 studies the conflict detection problem and proposes a microscopic event simulation model. Chapter 3 focuses on the conflict resolution problem proposing an extension of the alternative graph approach and various algorithms to optimize this novel model.

**Part II** is composed of Chapters 4, 5 and 6. Chapter 4 presents a prior study of the algorithms used in this part and describes a new framework for free derivative algorithm hybridization. Chapter 5 is focused on the description and calibration of a nested logit model used to represent the behaviour of the passengers depending on the timetable. In Chapter 6 the complete HSR model is defined based on the approach proposed in Chapter 5 and including a mesoscopic simulation model for representing train movement in the HSR network. The perspective desired is the maximizing of the benefits obtained by the railway operator.

Finally Chapter 7 gives the final conclusions and provides some lines for further research.
Part I: On-line approach
2. A discrete event-based simulation model for real-time traffic management in railways

PART I: ON-LINE

PART II: OFF-LINE
2 A discrete event-based simulation model for real-time traffic management in railways

Rail systems are highly complex and their control requires mathematical-computational tools. The main drawback of the models used to represent railway traffic, and to resolve any conflicts that occur, is the large computational time needed to obtain satisfactory results. Therefore the purpose of this Chapter is to study and design a discrete event-based model characterized by the positioning of trains in block sections, which can represent the rail system, including the dynamic aspect, and a fixed block signalling system able to pro-actively detect and resolve potential conflicts that may occur within this system. The aim is to reduce the computational cost as much as possible and implement the proposed model in a railway network. A numerical investigation, based on Renfe Cercanías Madrid rail network (Spain) shows the high computational performance on real sized applications of the proposed approach.

2.1 Introduction

Railway systems provide large-scale public transport services in an eco-friendly and sustainable manner. In order to increase performance and maintain a high quality service new strategies for increasing capacity are needed, including the building of new infrastructures and the improvement of existing ones. The improvement of these railway infrastructures requires heavy economic investment which will on occasions require limited changes to the existing infrastructure, which may not be possible for other reasons. Therefore, production-based strategies for increasing capacity by allowing more trains to be operated on the same infrastructure, or by train
CHAPTER 2. A DISCRETE EVENT-BASED SIMULATION MODEL FOR REAL-TIME TRAFFIC MANAGEMENT IN RAILWAYS

scheduling, are being developed.

Within the normal working of railway systems, they may suffer technical failures and disturbances, causing what are known as primary delays. The interaction between trains as a result of these primary delays may cause knock-on effects among other rail services, causing secondary delays. This problem is being addressed via an off-line approach, developing analytical methods for obtaining robust timetables that are able to deal with minor delays occurring in real-time, and on-line; by new technology which introduces Intelligent Transportation Systems (ITS); these allow railway operators to improve traffic control actions which reduce buffers between trains, and improve line capacity.

These ITS are designed as Decision Support Systems for dispatchers, and are divided into distinct modules. This type of design allows better maintenance and modification of the ITS since the adjustment of any of the components or the existence of any internal error will not affect the rest of the system; furthermore, any of the modules may be used individually. A prototype design/architecture based on ROMA D’Ariano and Pranzo (2009) is shown in Figure 2.1, in which three main modules may be observed:

Figure 2.1: ITS architecture

1. Load Data module. This module has the function of obtaining real-time system state data. It loads information on the Master Schedule and periodically verifies the state of the lines to check that the service schedules are being satisfied.

2. Railway Model module. The information obtained by the previous module is sent to the Railway Model module which assesses the current state of the system and the initial schedule with the aim of anticipating possible disruptions, secondary delays and problems which could occur on the railway system. This module is responsible for automatic traffic supervision.

2.2 Past research

will activate the Real-Time Optimization module which resolves possible conflicts and offers solutions which minimize as far as possible both the knock-on effects of delays and any other situation which may disrupt the system. This module is responsible for real-time train scheduling and routing. This module may also compute the optimal speed profile for each train D’Ariano et al. (2007b) Corman et al. (2009).

The ITS system assists the dispatcher in the taking of decisions, making real-time adjustments either to the original network schedule or the planned services, by cancelling trains, setting new routes, modifying speeds or by using other rail traffic control strategies.

Simulation models provide a detailed and dynamic description of the railway system and are the ideal analytical tool for representing the Railway Model module.

This Chapter concentrates on providing a computational assessment of the performance of a new simulation model for the real-time management of railway traffic representing the Railway Model module by an ITS as a part of the complete Decision Support System. The proposed model uses a simulation scheme for discrete events defined by the location and speed of the trains on the block sections (Flamini and Pacciarelli (2008)) and the signal controls.

The simulation model described in this Chapter is able to represent microscopically the working of existing rail systems with reduced computational cost in order to facilitate its future inclusion in an ITS, as shown in Figure 2.1. In this way it can also be used to predict the future state of the system, enabling possible disruptions to be anticipated and different options for system planning to be assessed.

The numerical experiments carried out on a real problem show the suitability of the model for on-line use, as well as other, off-line, uses, such as assessing alternative operation plans or emergency plans.

This Chapter is organized as follows. Section 2.2 discusses previous research, Section 2.3 proposes an improved simulation model based on discrete events, and presents the algorithm for solving the simulation model, in Section 2.4 the software application implemented (SIMEIFER) is presented and several computational experiments are reported, and finally Section 2.5 concludes with a discussion of our findings and future work.

2.2 Past research

Railway systems are highly complex and their planning and management require mathematical-computational tools. Traditionally a set of stages has
CHAPTER 2. A DISCRETE EVENT-BASED SIMULATION MODEL FOR REAL-TIME TRAFFIC MANAGEMENT IN RAILWAYS

been used, with different time horizons, which deal with certain types of established problems D'Aniano (2008).

A great deal of research is currently being carried out to find solutions to all these problems, with the aim of improving the quality of service provided. These approaches are generally divided into two areas, the off-line management of rail traffic, which focuses on the study of planning rail schedules from scratch as effectively as possible, and the real-time management of rail traffic, which deals with the solving of problems where there is already a schedule.

Where there is a predefined schedule although it may be highly robust Goverde (2005) and reliable Goverde and Hansen (2000) Landex et al. (2006), in unexpected cases or specific situations it is not able to respond appropriately to the changing needs of the system, nor to fulfil the original planning criteria Vromans et al. (2006), in which case it is necessary to find, within a time-window, a solution which can return the system to the scheduled state or to a state in which new disturbances are reduced as much as possible.

Simulation models have been applied in order to describe train movements. Depending on the level of detail required, there are two scheme types available: time-based, and event-based models. The first of these has a high computational cost and is applied to specific problems of energy consumption and signal layout design. With the second scheme it is possible to describe the movements of trains, such as arrivals and departures at and from stations and behaviour at intersections, as events. At the same time this kind of model can be classified according to the basic principles of signalling. While on long-distance and high-speed routes moving-block signalling is used, which is widely treated in the literature, on urban routes fixed-block signalling is more commonly used, and it is this system that will be the main focus of this Chapter.

Although the simulation uses algorithms of mathematical models, these simulations can also be combined with the real state of a system by data input/output when it is operating, which is useful for the real-time study of its behaviour and to make predictions about it.

The research in this area has led to the development of software tools which implement rail simulators with particular reference to long and middle-distance railways, such as RailSys Bendfeldt et al. (2000), OpenTrack Nash and Huerlimann (2004), FRISO Middelkoop and Loeve (2006) and TOPSIM Hellstrom et al. (2003). The differences between these simulators and the approach followed in this Chapter are that the proposed model is focused on urban railways, it offers users more precise modelling and visualization of infrastructure and is intended to be a conflict detection tool to launch the Real-Time Optimization module.

The differences between the model described in this Chapter and the solutions proposed by Dorfman and Medanic (2004) and Li et al. (2008) are
that they use a single track between stations. The proposed model, like the one proposed by Tornquist and Persson (2007), uses an approach aimed at solving the problem with several tracks on the same line, which gives a better representation of urban railways such as regional or underground trains Flamini and Pacciarelli (2008). It also includes a signalling system able to anticipate potential conflicts that may occur within this system, representing the real traffic lights used in railway infrastructures.

2.3 Formulation of the railway model

This section formulates a simulation model that represents the Railway Model module by an ITS, which can predict the future state of the system for a real situation and a given schedule. In order to formalize the railway model some definitions taken from D’Ariano et al. (2007a) will be used:

A railway system consists of a physical network which supports a network of services. This railway network comprises track segments and signals, the type changing from one country to another. The signalling system to be understood is the NS54 system, the standard used in many countries, including Spain. These signals are distributed along the length of the lines, intersections and stations, and define the block sections, which are track segments between two block signals.

The passing of a train through a particular block section is called an operation and the time required to carry it out is the running time. This time may experience a delay, depending on the entry signals for the train to the next track section. When a train leaves a track section a blocking period occurs until the entry of a new train is allowed; this is called the setup time. When two or more trains require the same block section at the same time a conflict occurs.

The model has been divided into three main parts (Figure 2.2):

1. Railway network model. The railway network is composed of track segments and signals which conform the block sections.

2. Service network model. This model represents the programmed services for a physical network.

3. Simulation algorithm. A simulation algorithm which represents the working of the entire system. That is, it represents the performing and sequencing of operations. In this way the instants in which operations will occur can be calculated and thus they can be compared with the original master schedule to identify any disruptions or delays.
2.3.1 Railway network

Physical layout

To represent the rail network in this model the typical make-up of urban railway networks, such as those of local or underground trains, has been borne in mind, in which each of the tracks to be used is divided into sections of different lengths called block sections. In each of these segments, for safety reasons, there must not be more than one train at a time. To avoid having more than one train enter these block sections a variety of blocking mechanisms may be used, of which the most common is the use of signals at the point where one segment ends and the next begins, which is the means used in this section.

Within the block sections themselves three main types can be distinguished, normal block sections, each with its individual restrictions, station block sections, which trains stops so that passengers may get on and off, and intersection block sections that represents the intersection zones. A further feature of these block sections is that they have a maximum speed \( V^i \) at which they may be crossed, which will depend on the physiognomy of the infrastructures or the state they find themselves in at a specific moment.

In summary, the main characteristics of the block sections will be the type (normal, station or intersection), the length of it, maximum speed and control traffic light state.

Thus, the physical network will be represented by a graph \( G = (V, A) \) where the \( V \) are the block sections, and the links will represent whether there is a connection between them. Furthermore, in the nodes there will be three types, \( V_N \) which are the normal block sections, \( V_E \) which are the station block sections, and \( V_I \) which are intersections, and so:

\[
V = V_N \cup V_E \cup V_I \quad V_i \cap V_j = \emptyset \quad i \neq j \quad i, j \in \{N, E, I\}
\] (2.1)
2.3. Formulation of the railway model

For example one part of the physical network of Renfe Cercanías Madrid (the local network) leads to a graph like that in Figure 2.3. It is possible to see the normal block sections can be seen in plain color, and the station block sections with hatching. This example does not include intersection block sections which are fictitious block sections representing the crossover points which exist in the network in more complex zones (Figure 2.12).

![Track Layout](image)

**Figure 2.3:** Example of node-link model of a railway track

Of the forms which exist to present a graph the format used in the L2QUE algorithm has been chosen Gallo and Pallotino (1988), which is used for directed graphs, since this way of storing data only uses two vectors of non-null elements and can quickly recover any type of information contained therein.

**The signalling system in the physical network**

In our model the signalling system works by means of fixed-block signalling. Each block section $j \in V$ will have a variable of state $S^j$ which expresses the
current state of the signal showing whether or not it may be entered.

The rules of signal control depend on the direction of travel of the trains with respect to the orientation of the tracks. In the normal orientation if a train is on a block section the entry signal to the track is on RED (R) preventing any other train from entering this block section, the entry signal to the adjacent track will be on YELLOW (Y) allowing entry to the block section but with certain precautions as to speed, and the signal to other tracks on GREEN (G) allowing entry without any restriction. An example of a possible situation of the system may be seen in Figure 2.4. This represents a change of block section for the train which was in the block section $V_i$ to a new block section $V_j$ and the changes in the signals. The lowercase letters $i$, $j$ and $k$ represent the identifier of the block section or the signal associated with this block section.

Therefore, when a train changes from a block section $V_i$ to another block section $V_j$ it will be necessary to update the signal $S_j$ to $R$ and, if it is occupied by another train, the signal $S_i$ to $Y$ and the signal $S_k$ at the previous block section $V_k$ to $G$.

![Figure 2.4: Example of the state of the signals at a particular moment on the C5 Line of Renfe Cercanías Madrid in the stations of San José de Valderas and Alcorcón](image)

Furthermore, in intersection block sections its signal remains at $R$ until the train leaves the next block section it reaches.

### 2.3.2 Service network

**Services and timetable**

In order to model the trains that operate on the available infrastructure and their associated characteristics the concept of services is introduced, which are individual options intended to satisfy the transport needs of the users of the rail system.

Each service comprises:
2.3. Formulation of the railway model

- A train with an associated maximum speed.
- A schedule to be kept to, defined by the time of departure and of passage through the stations.
- A route on the graph of block sections.

Let us suppose there are \( n \) services programmed on the rail network, collectively described as the set \( S \) and a set of \( m \) trains called \( T \) which will perform these services. Let \( t \) be an element of \( T \) and \( s \) an element of \( S \).

The timetable of each of the services \( (H_s) \) will be given by a vector whose coordinates are associated to the moments of departure, when the train leaves block section \( i \in V_E \) corresponding to a given station.

\[
H_s = (..., h_i, ...), i \in V_E \quad (2.2)
\]

Each service \( s \) has an associated route \( r_s \) in the block section graph \( G \). This sequence of block sections defines at each instant the next block section to advance to as a function of the current state of the train, and so define:

\[
R = \{ r_s | s \in S \} \quad (2.3)
\]

Thus a service \( s \) is determined by the triplet:

\[
s = \{ t_s, H_s, r_s \} \quad (2.4)
\]

where \( t_s \) is the train which performs the service \( s \).

**Rolling stock restrictions**

Different services share the same rolling stock, which means that a delay in certain services can affect the departures of subsequent services. These delays are managed by a variety of dispatching policies. This model considers that a service may not start until the train \( t \) which is due to perform the service is in the station, plus a turning buffer time between train routes D’Ariano and Pranzo (2009), which is a time margin between the end of a train route and the start of a new service using the same rolling stock.

Let us suppose that a train \( t \) performs two consecutive services \( s_1 \) and \( s_2 \) and that the planned timetables are:

\[
H_{s_1} = (h_0^1, h_1^1, ..., h_n^1) \quad \text{with} \quad h_0^1 < ... < h_n^1 \quad (2.5)
\]
CHAPTER 2. A DISCRETE EVENT-BASED SIMULATION MODEL FOR REAL-TIME TRAFFIC MANAGEMENT IN RAILWAYS

\[ H_{s_2} = (h^2_0, h^2_1, ..., h^2_m), \quad h^2_0 < ... < h^2_m \]  \hspace{1cm} (2.6)

Now:

\[ \hat{h}^2_0 = \max\{\hat{h}^1_n + TBT_{s_1, s_2}, h^2_0\} \]  \hspace{1cm} (2.7)

\[ \hat{h}^1_n \equiv \text{predicted finishing time of service } s_1 \text{ by train } t \]

\[ \hat{h}^2_0 \equiv \text{predicted starting time of service } s_2 \text{ by train } t \]

\[ TBT_{s_1, s_2} \equiv \text{turning buffer time between services } s_1 \text{ and } s_2 \]

2.3.3 Simulation algorithm

During train operation two situations are distinguished. The first is movement within a block section (operation) and the second is the transition from one block section to another. In the first type of movement the train does not interact with any other train, while in the second a conflict could be produced which would require a sequencing of operations involving several trains. In the following sections each of these movements is described.

Intra-block section movements (operations)

Suppose that an operation is performed in which a train \( t \) moves within a block section \( i \in V \) in the direction of block section \( j \in V \) as shown in Figure 2.5. The model assumes that the intra-block section movements of the trains are determined by the speeds and the time instant at which the train \( t \) occupied the start and finish of block section \( i \). Define the following variables:

- \( I^i_t \): The initial speed at entry for train \( t \) in block section \( i \).
- \( E^i_t \): The exit speed of train \( t \) on leaving block section \( i \) due to the restrictions imposed during the passage and the state of the controls.
- \( \gamma^i_t \): The instant at which train \( t \) enters block section \( i \).
- \( \tau^i_t \): The instant at which train \( t \) reaches the end of block section \( i \).

In this section is described the mechanisms for calculating the speed and the moment at which the train reaches the end of block section \( i \) as a function of the initial speed \( I^i_t \) and the instant \( \gamma^i_t \) at which it was at the start of the
2.3. Formulation of the railway model

block section, the state of the system and the characteristics of the block section.

From the previous characteristics three restrictions related to the propagation of speeds in the network as a function of the state of the system at a given moment are derived:

- The first of these is the conservation of speed when the train $t$ passes from block section $i$ to block section $j$ (Figure 2.5)

$$E_i^t = I_i^t$$ (2.8)

![Figure 2.5: Conservation of speed and instant when there is a change of block section](image)

- The second restriction is the speed as a function of signal regulation $S_j$ of block section $j$ which it is about to enter

$$E_i^t := \begin{cases} V_{i,j}^{max} & \text{if } S_j = G \\ V_Y & \text{if } S_j = Y \\ 0 & \text{if } S_j = R \end{cases}$$ (2.9)

where $S_j$ is the signal of block section $j$, $S_j \in \{G, Y, R\}$ are the different states, $V_{i,j}^{max}$ is the maximum speed associated to the train $t$ and $V_Y$ a safe speed predetermined by the system managers.

It is assumed that train $t$ travels in block section $i$ at a mean speed $M_i^t$:

$$M_i^t = \frac{I_i^t + E_i^t}{2}$$ (2.10)

- A special type of block section is considered which can limit the speed to a value $V^i$ which cannot be surpassed at any time for safety reasons, and so if $\max \{I_i^t, E_i^t\} \geq V^i$ then $E_i^t = \min \{M_i^t, V^i\}$ and $M_i^t = V^i$.

The running time, that is, the time taken to traverse a complete block section $C_i^t$ is

$$C_i^t = \frac{L_i}{M_i^t}$$ (2.11)
where $L_i$ is the length of the block section. Note that this section is concerned here with urban networks, basically used for transportation between the different zones of a city and its outskirts, where the length of block sections is small, and so formulae (10) and (11) provide a good approximation.

Finally, two characteristics related to the stops made in station block sections with a platform where passengers may board and alight have also been considered:

- To the running time $C_i^t$ must be added a time $T_p$ defined as the time necessary for passengers to get on and off the train, in accordance with the safety regulations, plus the time required to carry out braking and accelerating manoeuvres, which refers to the reduction of speed when entering a station until the train stops and the acceleration after the train has picked up all the passengers from the platform. $T_p = 0$ if the block section is not of the station type.

- Bearing in mind that there is a schedule $H_s$ it is assumed that the trains can at no point be ahead of their scheduled time, that is, they cannot be in advance of their designated time for block section $i$ given by $h_i$. This case could be obviated by following a different approach, because letting some trains arrive earlier at some points, if carefully managed, can be used to optimize railway traffic D’Ariano et al. (2008b).

For these reasons the time of arrival $\tau_i^t$ at the end of block section $i$ is given by

$$\tau_i^t = \max \{ \gamma_i^t + C_i^t + T_p, h_i \}$$

(2.12)

where $\gamma_i^t$ is the instant of entry to block section $i$. $\gamma_i^t = \tau_i^k$ where $k$ is the block section previous to $i$.

It is necessary to bear in mind that not only speed is conserved at the change of block section, but also the instant, that is $\tau_i^t = \gamma_i^t$.

The mechanisms described above correspond to the propagation of speeds and instants in the unions of block sections. In order for the problem to be completely determined only the initial conditions are missing. Assume that $I_i^t = 0$ for the initial stations $i$ of a service and $\tau_i^t = \hat{h}_i$, that is, the real end time of the service performed by train $t$.

**Inter-block section movements (events)**

An event in the system is defined as a transition in the state of the system in which a service $s$ leaves a block section $i \in V$ in which it was travelling and
2.3. Formulation of the railway model

enters another block section \( j \in V \). These events coincide with the different operations carried out in the system.

The key to performing the simulation is to analyse the events in the appropriate order since they are interrelated and a movement of one train can affect the schedule of another. In other words, the space (block sections) and time in which the events occur must be appropriately analysed. The dispatching rule used is the First Come First Served (FCFS) which is commonly adopted in railway management. It consists of giving precedence to the train arriving first at a block section, and sequencing the trains in order of arrival.

Other possibilities are to use the First Leave First Served (FLFS) rule or the AMCC algorithm described in D’Ariano et al. (2007a). The FLFS dispatching rule consists in giving precedence to the train which is able to leave the block section first, computing this time before entering it. The AMCC algorithm is a greedy heuristic algorithm based on local information which uses the concept of alternative graphs. Its objective is to compute near-optimal train schedules of practical size within a short computation time selecting in each iteration one of two alternative arcs which then fixes a precedence constraint between two trains at a potential conflict point.

Suppose that it is wished to simulate the system in the time interval \([T_l, T_f]\). Let \( E \) be the set of events which occur in the system in the interval \([T_l, T_f]\). For each event \( e \in E \) the associated block section \( i \in V \) the time \( \tau_i^e \) at which event \( e \) should occur, and the service \( s_e \) associated with event \( e \), must be stored.

It may be seen that each event is associated with a train \( t \in T \) and therefore at each moment the system state is completely determined by the distribution in block sections of this set of trains. This will allow the list of events \( E \) to be processed in an orderly manner in accordance with the FCFS dispatching rule. Suppose that at the current time \( T \), the train \( t \) is in block section \( i \in V \). Then set \( Q_t = \tau_i^t \).

The set of instants at which the events that determine the state of the system occur is, at time \( T \):

\[
Q = \{ Q_t \mid Q_t > T, t \in T \} \tag{2.13}
\]

The next instant at which the system may change state will be

\[
Q'_t = min_{t \in T} \{ Q_t \} \tag{2.14}
\]

Note that the state of the system is unaltered in \([T, Q'_t]\) and so the system clock (the time within the simulation) can be reset by \( T = Q'_t \). Then, once the train \( t' \), its block section and service have been identified, it is, if possible,
moved to the next block section by the transition proper to it according to the intra-block section mechanisms previously described, and the new $Q_t'$ is calculated.

A conflict may occur, and therefore the block section $j$ which the train $t'$ seeks to enter is already occupied by another train $t''$ which has not kept to the schedule due to some type of unforeseen circumstance, and so it must wait until the block section is free. The minimum instant at which train $t'$ may leave the block section is $Q_t' = Q_t'' + \epsilon$ where $\epsilon > 0$ is the minimum established safety time known as the setup time.

The algorithm used for working with the list of events $E$ should update the current state and obtain the new situations which will occur according to all the considerations in the model. This algorithm is described in Algorithm 1.

---

**Algorithm 1** of Simulation of Discrete Events in a Railway System

**Input:** Graph of block sections $G = (V, A)$, set of programmed services $S$, timetable sheets $H_s$, $s \in S$, list of trains $T$ and end time of simulation $T_E$  

(Initialization). Make list of current events $E$ from the set of initial departures of $T$. Let $T_I$ be the initial time of the simulation. Make $T = T_I$

while $T \leq T_E$ do

(Choose train associated with event to be handled) $t' = \arg \min_{t \in T} \{Q_t\}$

$T = Q_{t'}$

(Handle event) From $t'$ calculate the next block section $j \in V$ for entry of service $s$ from its route $r_s$.

if $t'$ has reached the final station of the service then

Eliminate event of $E$

Generate the next service $s$ and event for train $t'$ if it exists taking into account restrictions on rolling stock.

else

Calculate time of the next operation of train $t'$ $Q_{t'}$ according to the criteria described for intra-block section movements.

if $S_j = R$ then

$Q_{t'} = \max(Q_{t'}, Q_{t''}) + \epsilon$ where $Q_{t''}$ is the instant of departure of the train in block section $j$ at that moment and $\epsilon$ is the setup time.

end if

(Update signals). Update affected signals and new block section $j$ which train $t'$ is now in.

end if

end while

(End) View results of simulation.
2.4 Computational experiments

This section works through a computational experiment with the simulation model described in a real case.

Two numerical tests have been designed. In the first the aim is to test whether the CPU time is lowered enough for the model to be applicable online. In the second test an off-line use is presented in which it is assessed, that is, integrating into an ITS for study a new use or operational plan considering the number of conflicts that occurs.

To carry out the computational experiments the SIMEIFER tool was developed following the United Software Development Process to implement the model mentioned in the previous section. The tool has been codified using MATLAB. This tool provides an easy to use visual editor of railway infrastructure, and allows input and editing of the trains and services which will participate in the simulation, or they may be acquired from databases in which the planning schedule is stored, and it can show and store the results of any simulation that has been performed. Some videos showing the working of this tool may be seen at links of Appendix B.

Figure 2.6 shows a capture of the simulation tool. Area 1 represents the part the user is focusing on the railway system at the moment from area 2, which shows the whole infrastructure. Areas 3 and 4 are the parts of the system that handle the information about trains and services respectively, including options to add or load trains or services to the system.

The computer used to perform the computational experiments has the following characteristics: Windows Vista 64 bits, processor: Intel QuadCore 2.83 GHz, RAM: 4GiB.

2.4.1 Case study of Line C5

To check the applicability of the simulation model several computational experiments have been performed on the Line C5 of Renfe Cercanías Madrid (Spain), which may be seen in Figure 2.7. This line is formed by a set of 256 block sections on which there are 23 stations over 45 km of track, carrying more than 400,000 passengers every day. It also has sensors which can show the complete state of the system at any time. This example uses information provided by Renfe Cercanías which gives the schedules of the 327 trains which operate daily on the line.

The first scenario to which the model is applied corresponds to a normal running of the system with initial timetables predefined, without including any disruption or disturbance.

The second scenario deals with one of the problems which is most
CHAPTER 2. A DISCRETE EVENT-BASED SIMULATION MODEL FOR REAL-TIME TRAFFIC MANAGEMENT IN RAILWAYS

Figure 2.6: The SIMEIFER tool

frequently encountered in railway systems, the closing of a set of block sections because of some kind of emergency or line work. An example of this occurred on the 26th June 2009 on the Line C5 in Madrid, when a crash between two trains cut the links between Madrid-Atocha and Móstoles el Soto.

Using SIMEIFER it is possible to redirect the traffic automatically, removing the obstructed block sections from the infrastructure and advice could be obtained as to whether it is feasible to redirect the traffic through the available lines without causing further accidents or bringing the system to a halt.

In order to carry out this simulation part of Laguna station on Line C5 has been removed, as can be seen in Figure 2.8.

In this case study the following results have been obtained:

- Simulated Time (ST). Shows the total simulated time, that is, the time taken by the system to complete the given schedule.

- Simulation Time (sT). Shows the total CPU time taken to perform the complete simulation of the given schedule.

- Waiting Time (WT). Sum of all waiting times caused by trains which have met with a yellow or red signal.
2.4. Computational experiments

Figure 2.7: Line C5 of Renfe Cercanías Madrid

Figure 2.8: Modification to Laguna station

- Delay Time (DT). Sum of all the delays experienced by the trains in leaving a station later than the time established in the initial schedule.

The results for both scenarios are presented in Table 2.1. Figure 2.9 shows part of the time-station diagram for the first scenario.
CHAPTER 2. A DISCRETE EVENT-BASED SIMULATION MODEL FOR REAL-TIME TRAFFIC MANAGEMENT IN RAILWAYS

<table>
<thead>
<tr>
<th>Scenario</th>
<th>ST</th>
<th>sT</th>
<th>WT</th>
<th>DT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal</td>
<td>24h 16min 40sec</td>
<td>7min 43sec</td>
<td>32min 33sec</td>
<td>36min 21sec</td>
</tr>
<tr>
<td>Emergency</td>
<td>24h 40min 26sec</td>
<td>7min 56sec</td>
<td>2h 43min 46sec</td>
<td>3h 8min 57sec</td>
</tr>
</tbody>
</table>

Table 2.1: Results of the simulation for the standard timetable on Line C5

Figure 2.9: Time-Station diagram for first scenario

It can be seen that for the first scenario the waiting times and system delays are small enough to be applied in real-time. There is an average waiting time of 5.97 seconds for each train of the 327 planned, and an average delay time of 6.67 seconds per train, which, considering all the stations on the routes, is insignificant, especially if it is also taken into account that it must be divided by the number of block sections and stations which form part of the route.

The numerical experiments carried out with the network parametrization provided by Renfe Cercanías with regard to average journey speeds, maximum speeds, length of block sections, time stopped at stations etc, show clearly how the model described can reliably simulate a real railway system, getting the trains to pass through the stations at the intended moment, since, in theory, the schedule produced by Renfe Cercanías Madrid should give, for each train in each station or block section, a waiting time or delay time similar to those given by the model since it reproduces the planned schedules very precisely, with an average delay of only 6.67 seconds for an average journey of 55 minutes.

In the results for the second scenario a general increase in the waiting times and delay times can be seen with respect to the first, but even so it is clear, once the simulation has finished correctly, that it would be possible to use the sole remaining part of the station to perform transfers in both directions. It can also be observed that the simulation time is small, and in less than 8 minutes results could be obtained that would help the dispatchers to take the most appropriate decisions in this situation.
2.4. Computational experiments

Finally, Figure 2.10 shows the evolution of the computational cost with respect to the instant simulated for the first scenario; the cost is highest at the 500th minute, which roughly coincides with the morning rush hour, about 8:30 am. This graph shows on the OY axis the number of seconds required to simulate 10 minutes (600 seconds) of the system at each instant \( T \) of the simulation. Note that even in the rush hour this figure is less than 12 seconds, giving a ratio of \( \frac{600}{12} = 50 \) which shows that the system can calculate 50 times quicker than real-time even in the worst situation.

![Figure 2.10: Graph of the running of the base example](image)

This section will now look in more detail at the matter of whether the proposed simulation model can be included in an ITS to anticipate situations and as a means of predicting the future state of the system.

The on-line applicability of an ITS system like that shown in Figure 2.1 requires the following tasks to be performed:

- **Load Data (LD).** Load system information in real-time.
- **Conflict Detection (CD).** Run the *Railway Model module* to detect possible disturbances and if necessary run the *Real-Time Optimization module* and inform the human dispatcher.
- **Operation Optimization (OPT).** Run the *Real-Time Optimization module* to resolve conflicts. The cost of running it will depend on the algorithm used.

The computational cost of carrying out the tasks \( LD, CD \) and \( OPT \) depends on a variety of factors, of the most important is the time horizon of the operations to be studied and the trains in service during that time.
window. The time horizon of practical interest to railway managers is usually less than an hour for real-time purposes.

A number of numerical trials have been performed on the Line C5 with a time horizon of 1 hour, and the mean value obtained for the time necessary to load the data for the time window was \( LD = 57.23 \) seconds. D’Ariano et al. (2007a) reports mean computational results obtained with AMCC algorithm (element Real-Time Optimization module) for networks of similar size to the Line C5 and with a time window of 1 hour as \( OPT = 0.5 \) seconds.

Figure 2.11 shows the estimated computational cost of running the ITS at each moment of time. If the system is refreshed every 10 minutes it can be seen that in the worst case the dispatcher would have 7 minutes and 54 seconds to take a decision and to communicate the necessary measures to resolve the disturbance.

\[ \text{Figure 2.11: Computational cost of ITS with time horizon 1 hour every 10 minutes} \]

### 2.4.2 Study of alternative operation plans

Finally, another use for the model is to study possible future actions, such as the extension of the network by building new track sections, stations or entry points, modernizing or improvement of infrastructures or increasing services.

The choice of a plan of use from among a set of alternatives requires assessment from many perspectives (operating cost, demand, etc) but also of the impact on current use. SIMEIFER allows this assessment.

One of the most important actions currently being undertaken by Renfe Cercanías Madrid consists of the building of new lines to Barajas airport
2.4. Computational experiments

from the station of Chamartín and three stations, Manoteras, Valdebebas and Barajas T-4 on the route. This requires the study of alternative operation plans for the new, modified, system, seeking the lowest number of conflicts and the shortest waiting times.

Specifically three alternative operation plans will be studied in which the intention is to put 475 trains into operation, and the aim is to find an optimal solution without greatly modifying the current plan of the system, and with a minimum number of conflicts at intersections (Figure 2.12).

In this case study the following results have been obtained, as well as the parameters ST, sT and WT mentioned in the previous section:

- Conflict Number (CN). The total number of occasions on which the trains in the system have had to wait on meeting a red signal.

- Time between Conflicts (CT). The average time in the system between one conflict and the next. This parameter measures the complexity that the dispatchers have to deal with.

The results for the simulation of the three alternatives using our model are shown in Table 2.2 and Figure 2.13.

![Figure 2.12: Example of crossover zones](image)

These results clearly show the difference which exists between the three alternatives considered for the system plan, both in terms of waiting times produced by conflicts and the number of conflicts produced, particularly between 8:00 and 9:00 o’clock. These results are in agreement with the experience of the dispatchers, and allow a precise assessment of the viability of the different alternatives by using the model.
CHAPTER 2. A DISCRETE EVENT-BASED SIMULATION MODEL FOR REAL-TIME TRAFFIC MANAGEMENT IN RAILWAYS

<table>
<thead>
<tr>
<th>Plan</th>
<th>ST</th>
<th>sT</th>
<th>WT</th>
<th>CN</th>
<th>CT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>19h 36min 32sec</td>
<td>3min 33sec</td>
<td>2h 49min 7sec</td>
<td>110</td>
<td>11min</td>
</tr>
<tr>
<td>2</td>
<td>19h 23min</td>
<td>2min 27sec</td>
<td>1h 30min 44sec</td>
<td>57</td>
<td>21min</td>
</tr>
<tr>
<td>3</td>
<td>19h 23min</td>
<td>2min 3sec</td>
<td>56min 56sec</td>
<td>30</td>
<td>39min</td>
</tr>
</tbody>
</table>

Table 2.2: Results of simulation for different plans

![Conflicts/hour for alternative operation plans](image)

Figure 2.13: Conflicts/hour for alternative operation plans

2.5 Conclusions

In this Chapter has been described a simulation model for a railway system based on discrete events determined by the position of the trains on the different block sections which have mechanisms for signal regulation. This simulation tool can be integrated into on-line or off-line systems. In the first case it would form part of an ITS and would allow the prediction of the future state of the system for a real situation and a given schedule, and would thus anticipate possible conflicts and secondary delays. Off-line use is aimed at assessing and testing the feasibility of operation plans determined by a set of services and timetables.

The software tool SIMEIFER has been developed from this model, and it satisfies the requirements and offers the features which would be expected of a system of this nature. The tool has been validated by the performance of several experiments on the real situation of the Line C5 of Renfe Cercanías Madrid and on the development plans currently under consideration to
2.5. Conclusions

connect the station of Chamartin with Madrid-Barajas airport. It has been shown that the computing time required to assess the evolution for the next 10 minutes takes 12 seconds in the most difficult case and an average of 5 seconds, which proves its viability for on-line use. In the numerical trials off-line use of the model for the assessment expansion plans for the rail system has also been demonstrated.

One of the essential characteristics of the ITS's developed for the control and management of rail traffic is their modular architecture, such that they allow the incorporation and combination of different modules capable of solving any of the difficulties involved in railway systems. For this reason Chapter 3 addresses the implementation of the *Railway Module* within a real system, with the aim of detecting disruptions which might occur during the running of the initial plan, giving a useful system for providing strategies for conflict resolution such as FCFS, the AMCC algorithm D’Ariano et al. (2007a) and a new approach which allow the problem of real-time conflict resolution to be addressed.
3. A demand-based weighted train delay approach for rescheduling railway networks in real time
A demand-based weighted train delay approach for rescheduling railway networks in real time

Rail systems are highly complex and their control in real-time requires mathematical-computational tools. The main aim of these tools is to perform swift optimal rescheduling in response to disruptions or delays caused by events not foreseen in the original plans, so that there is no knock-on effect on other services due to these primary delays. This Chapter proposes a novel demand-based weighted train delay approach based on the alternative graph concept for rescheduling passenger train services which can be implemented as a Real-Optimization module in the ITS architecture defined in Chapter 2. This problem is formulated as a binary integer linear programming problem which tries to maximize consumer satisfaction by minimizing total passenger delay at destinations. A heuristic method, the so-called Avoid Most Delayed Alternative Arc (AMDAA) algorithm, is proposed to solve the model. AMDAA is an adaptation of Avoid Maximum Current Cmax (AMCC) to the new model. A numerical comparison is carried out with AMDAA, a Branch-and-Cut method, AMCC and the heuristic First Come First Served (FCFS). Numerical research carried out with data from the Renfe Cercanias Madrid rail network (Spain) shows the high computational performance in real applications of the algorithms and the suitability of this weighted train delay based on demand model versus the classical makespan minimization approach.

3.1 Introduction

Currently, railway systems, both of goods and of passengers, represent one of most heavily used forms of public transport in developed countries, and demand is constantly increasing. Rail networks are expanding and growing.
CHAPTER 3. A DEMAND-BASED WEIGHTED TRAIN DELAY APPROACH FOR RESCHEDULING RAILWAY NETWORKS IN REAL TIME

creating very large, complex systems.

The railway rescheduling problem has been defined in an on-line context without uncertainty, to try to handle an unexpected disruption that occurs in real-time. Kraay and Harker (1995), Narayanaswami and Rangaraj (2012). This situation may lead the railway system to be incapable of properly addressing the requirements of the system or satisfying the original timetable and requires network recovery to be achieved in a short time. Recovery can be carried out attempting to return train to their original timetables or generating new temporary timetables for the remainder of their journeys.

In literature about train timetabling problem, the demand is a key factor and it is exhaustively considered Cacchiani and Toth (2012). On the other hand, in rescheduling the existence of a passenger dissatisfaction-based timetable is assumed so its main objective is to recover the system to the pre-established timetable, not directly considering demand in this phase. In this Chapter is assumed that demand data is available and can be taken into account in the process of recovering the railway system.

The Chapter is organized as follows. Section 3.2 discusses previous research in rescheduling, Section 3.3 defines the concept of alternative graphs and the new approach proposed, Section 3.4 presents the heuristic algorithm used for solving the rescheduling problem, in Section 3.5 several computational experiments are reported to compare these solutions, and finally Section 3.6 concludes with a discussion of our findings and future work.

3.2 Past research

Many studies are currently being carried out to solve problems related to railways, we refer the reader to Caprara et al. (2007), Caprara et al. (2002) and Cordeau et al. (1998) for surveys in railway optimization. Focussing on the train-conflict resolution problem with the aim of improving the quality of the service offered, the main strategies followed are train timetabling, train dispatching, train platforming and train routing problems, Lusby et al. (2011).

This Chapter is focused on train-conflict resolution in real-time.

A mathematical tool widely used for modelling the TS problem is the so-called alternative graphs method described by Mascis and Pacciarelli (2002). These graphs represent the feasible moves for an individual train in time and space as nodes and fixed arcs and the conflicting train paths as a pair of alternative arcs. Any solution of the graph requires that one of each alternative arc pair be selected. This approach represents the train timetabling problem as job-shop scheduling, where the railway scenario is analogous to a shop with blocking and no-waiting behaviour. This
formulation will lead to a minimization of the makespan of trip times in a railway context.

The main difference with the alternative graph compared to other approaches presented in the literature is the detailed but flexible representation of the network topology with regard to railway signals and operational rules. This approach can easily incorporate a number of traffic regulation rules and constraints relevant to railways, which are rarely taken into account in the literature, as observed by D’Ariano (2008).

Mascis and Pacciarelli (2002) proposes a heuristic algorithm that tries to reduce the computational costs of solving the complete alternative graph called Avoid Maximum Current Cmax (AMCC). This algorithm compares in each iteration two alternative arcs and avoids the arc whose selection would result in the worst solution based on the evaluation of each path.

Mazzarello and Ottaviani (2007) apply this formulation for dynamic rescheduling after delays, minimizing delays and fuel consumption. Furthermore D’Ariano et al. (2007a) apply this concept to a rescheduling problem improving AMCC algorithm with the inclusion of the concept of static implications. D’Ariano et al. (2008a), Corman et al. (2010) and Corman et al. (2011b) use alternative graphs for re-routing trains in real-time. D’Ariano et al. (2008b) test the same model for dynamic timetabling for dispatching support. It is also possible to compute the optimal speed profile for each train using this model D’Ariano et al. (2007b), Corman et al. (2009). Another approach of rescheduling presented by Corman et al. (2011a) is to modify the objective function of the model including classes of priority for the trains.

Corman et al. (2012b) deals with the coordination of multiple regional control centres. These authors demonstrate that the coordination problem can be ideally solved with a branch and bound procedure.

Tornquist Krasemann (2012) detects that for certain scenarios it is difficult to find good solutions within seconds using a Branch-and-Cut approach. This paper proposes a greedy algorithm which effectively delivers good solutions within the permitted time.

Currently a growing interest exists in how to represent the demand decisions, leading to the development of models that combine the TS and DM problems.

Dollevoet et al. (2009) and Schachtebeck and Schöbel (2007) propose to use an approach aimed at minimizing the sum of all passenger delays plus the sum of all missed connections. Schachtebeck and Schöbel (2007) add capacity constraints to the delay management formulation.

Corman et al. (2012a) describes a bi-objective TS to minimize both the delay of the trains and the number of missed connections.

Kanai et al. (2011) deals with DM and TS problems combining simulation
and optimization. The simulation part consists of a train traffic simulator and a microscopic passenger flow simulator which traces the behaviour of passengers one by one. The optimization approach minimizes passenger dissatisfaction.

Almodóvar and García-Ródenas (2013) proposes a model for timetable rescheduling in emergency cases, reallocating trains/buses in real-time to other service lines. This model assumes that passengers use travel strategies and waiting passengers are loaded at trains/buses on a first-come-first-served basis. The infrastructure restrictions are not taken into account by the model.

Dollevoet et al. (2012) presents an integrated approach of DM and TS models. It determines which connections to maintain and proposes the departure and arrival times of trains at stations using a microscopic model.

Cadarso et al. (2013) proposes a two-step approach that combines an integrated optimization model for representing the timetable and the rolling stock restrictions with a multinomial logit model which simulates passenger behaviour.

Wang et al. (2013) shows a different approach to how to consider the demand, in which the objective function is a trade-off between energy consumption and comfort in transit.

3.3 Problem formulation

3.3.1 Alternative graph

A railway network comprises normal tracks and station tracks which include platforms. The tracks are divided into several block sections which are delimited by block signals.

Train movements are controlled by a signalling system which ensures that each block section may host only one train at a time. The entrance times of all trains within block sections must be synchronized to avoid conflicts. Traversing a block section by a train is called an operation and the planning of the movements of trains requires determining the start time of the operations.

A modelling approach presented in the literature by Mascis and Pacciarelli (2002) is the alternative graph model. This model turns out to be a powerful tool for scheduling problems, since it was designed to deal with a wide range of applications in which the response times are a critical factor in the assessment of the goodness of the solution. It also allows the inclusion in the model of all the relevant physical characteristics and the restrictions
3.3. Problem formulation

on the railway network in order to create efficient and realistic plans.

The scheduling problem is defined by the alternative graph formed by the triplet \( G = (N, F, A) \). This triplet comprises the set of nodes \( N \), a set of fixed directed arcs \( F \) and a set of alternative directed arcs \( A \).

- The set \( N \) comprises all the nodes of the graph. A node corresponds to an operation. Also, to include all the necessary information in the model, there are special types of node:
  - **Start node.** This is a dummy source node denoted by 0.
  - **Finish node.** A dummy sink node denoted by \( n \), where \( n - 1 \) is the number of real nodes present in the system.
  - **Entry node.** For each train an entry node is added which represents the entry of a train into the system. It is assumed that the start node is connected to each entry node by a fixed arc labelled with the entry time of its train.
  - **Exit node.** This node represents the train leaving the system, all the exit nodes are connected by a fixed arc to the finish node. This arc contains the expected exit time of the train from the system. This set of nodes is denoted by \( N_{\text{exit}} \).
  - **Rest of nodes.** These indicates the current position (track segment) of a train.

The decision variables are the times at which the operations start, that is, the time at which a train enters a given block section. The start time of a generic operation \( i \) is denoted by \( t_i \), where \( i=1, \ldots, n \). The variable \( t_n \) indicates the moment at which all the trains have finished their routes.

- The set \( F \) is formed by fixed directed arcs \((i, j)\), which represent a relationship of precedence, forcing the start time \( t_i \) of operation \( i \) before the start time \( t_j \) of operation \( j \). The set \( F \) defines the sequence of operations carried out by the trains. If a time \( f_{ij} \) is required by the operation \( i \) and \((i, j) \in F \) thus, \( t_j \geq t_i + f_{ij} \).

The set \( F \) also includes the arcs which join the start node with the respective entry nodes to the system of each train and its weight is associated to the entry time of the train to the system, as well as the arcs which join the exit nodes of each train with the finish node and its weight is associated to the predicted leaving time of the system.

- **Alternative arcs** are contained in the set \( A \). A pair of alternative arcs represents the precedence between two operations on the same resource, in this case a block section. Given a pair of arcs \(((i, j), (h, k)) \in A \), arcs \((i, j)\) and \((h, k)\) are said to be paired and arc \((i, j)\) is the alternative of arc \((h, k)\). Arcs in this set model time constraints between operations of conflicting jobs and selecting one of the arcs that belongs to a pair of alternative arcs over the other enforces the sequencing and timing decision, so the selections of alternative arcs are the decision
variables of the problem. Their weight corresponds with the setup time that ensures a minimum headway between consecutive trains.

This approach considers the idea that the logical order of tasks is related to the defined schedule, thus the predefined route of each train is what defines the order of each task \( i \). This constraint is included in the creation of the fixed arc set chaining fixed arcs from node to node, and the MIP formulation defines a constraint which forces the start time \( t_i \) of operation \( i \) to be before the start time \( t_j \) of operation \( j \). If a time \( f_{ij} \) (travel time) is required by the operation, then \( t_j \geq t_i + f_{ij} \). That is, the route of each train is implicitly defined by set \( F \).

Therefore this model is a microscopic model which defines each movement in each track segment, so it does not matter if is a double-tracked line or another network configuration. The most important thing is to define the route of each train completely. The proposed formulation and approach does not focus specifically on conflict detection and resolutions in stations but allows the formulation of scheduled stops, rolling stock connection, passenger connection and route booking constraints D’Ariano (2008).

Figure 3.1 shows an example of a railway network with three trains. The trains will request the use of track segments 3 and 4 before actually occupying them. While traversing a sequence of track sections 3 and 4, trains will successively release each one once the tail of the train has exited, and a small setup time has elapsed. Released track sections may then be claimed by other trains. Figure 3.1 (at bottom) shows its alternative graph to model these precedences, in which fixed arcs are represented in black with continuous lines and each pair of alternative arcs is in colour with discontinuous lines. Alternative arcs model the possible sequencing decisions between trains 1 and 2 on track segment 3 and the precedence relationship in track segment 4 between all the trains.

The classical formulation finds a feasible schedule which minimizes the so-called makespan

\[
t_n - t_0 = \max_{j \in N_{exit}} \{t_j\} - t_0
\]

where the times \( t_j \) with \( j \in N_{exit} \) represent the finishing time of the last operation of the set of trains.

This problem is formulated as an integer linear problem with disjunctive constraints:

\[
\begin{align*}
\text{Minimize} \quad & Z_{\text{MAKESPAN}} = t_n - t_0 \\
\text{subject to:} \quad & t_j - t_i \geq f_{ij}; \quad (i,j) \in F \\
& (t_j - t_i \geq f_{ij}) \lor (t_h - t_k \geq f_{hk}); \quad ((i,j),(h,k)) \in A
\end{align*}
\] (3.1)
3.3. Problem formulation

3.3.2 A weighted train delay based on demand approach

The makespan approach described in the previous section attempts to minimize the total operation time in the system, by calculating the overall time and subtracting the starting time of the operations.

This model is used to solve job-shop problems and it is suitable when the objective is to finalize a project (minimizing the final moment $t_n$) but is open to question in the management of rail traffic. If a train experiences a delay and arrives at a different intermediate point later than the moment planned, the passengers who are in this station will suffer a delay which could not be mitigated even though the train regained its original timetable. Also in a case where the schedule has been disrupted in various services at same time, this approach will only focus on the train with the biggest delay which could mean the system not taking into account services with minor delays.

The approach proposed in this Chapter tries to address these drawbacks. The main idea is to use the concept of alternative graphs presented below, but changing the objective function, focusing on minimizing the sum of delay accumulated during arrival at the stations and including the demand, seeking to reduce the overall time of passenger delay produced in the infrastructure.

It is assumed that $N_{\text{stop}}$ is the set of operations associated with scheduled stops in stations (i.e. in block sections) and $g_i$ with $i \in N_{\text{stop}}$ is an estimation
of the number of passengers that leave the train in the station \( i \) represented by an origin-destination (OD) matrix disaggregated by train services. Liu et al. (2010) present a review of methods used to obtain this data. These OD-matrices are frequently used in the Delay Management problem in an off-line context and for the rescheduling problem new technologies are required to calculate these estimations in real-time. These off-line estimations could be assumed to be the values of \( g_i \). However, if the OD matrix is not available it is possible to set all the values \( g_i \) to \( 1 \) which focuses only on train delay minimization at stations.

Moreover, it is assumed that the problem is posed in a rescheduling context and that the instants in which operations are scheduled \( \hat{t}_i \) with \( i \in N_{STOP} \) are known.

Kraay and Harker (1995) propose a nonlinear objective of delays at arrivals and departures for MISLP, each individually priced.

\[
Z = \sum_{i \in N_{STOP}} \beta_i |t_i - \hat{t}_i|^{\phi} + \sum_{i \in N_{DEPARTURE-STATION}} \alpha_i |t_i - \hat{t}_i|^{\phi}
\]  

(3.2)  

where \( N_{DEPARTURE-STATION} \) is the set of operations associated with the departure time for trains from stations.

This goal appears in cargo trains and it minimizes the costs emanating from the actual deviation of planned train schedules from real delivery time. These costs are delivery penalty and inventory costs in intermediate stations. From the point of view of the passengers only the delay at destination is relevant. For this reason the proposed objective function is:

\[
Z = \sum_{i \in N_{STOP}} g_i \max\{0, t_i - \hat{t}_i\}^{\phi}
\]  

(3.3)  

The parameter \( \phi \) has been set to 1 in order to address the computationally issue. The proposed optimization model is formulated as follows:

Minimize \( Z_{\text{delay}} = \sum_{i \in N_{STOP}} g_i \max\{0, t_i - \hat{t}_i\} \)  

subject to:

\[
t_j - t_i \geq f_{ij};
(t_j - t_i \geq f_{ij}) \lor (t_k - t_h \geq f_{hk});
(i, j) \in F
((i, j), (h, k)) \in A
\]

(3.4)  

The objective function of problem (3.4) is still related to train delay minimization, even if weighted with the number of passengers per train. It can be reformulated as the following binary integer linear programming
3.3. Problem formulation

problem:

\[ \begin{align*}
\text{Minimize} & \quad Z_{\text{MIN}} = \sum_{i \in N_{\text{STOP}}} g_i d_i \\
\text{subject to:} & \quad t_i - \hat{t}_i \leq d_i; \quad i \in N_{\text{STOP}} \\
& \quad t_j - t_i \geq f_{ij}; \quad (i, j) \in F \\
& \quad t_j - t_i \geq f_{ij} - M(1 - y_{ij}); \quad ((i, j), (h, k)) \in A \\
& \quad t_k - t_h \geq f_{hk} - M y_{ij}; \quad ((i, j), (h, k)) \in A \\
& \quad d_i \geq 0; \quad i \in N_{\text{STOP}}; \\
& \quad y_{ij} \in \{0, 1\}; \quad ((i, j), (h, k)) \in A \\
\end{align*} \] (3.5)

**Theorem 3.3.1.** Let \( M = \sum_{(i, j) \in F} f_{ij} + 2 \sum_{((i, j), (h, k)) \in A} (f_{ij} + f_{hk}) \), thus the linear integer programming problem (3.5) is equivalent to (3.4).

**Proof.** We begin by showing that the variable \( d_i \) for all \( i \in N_{\text{STOP}} \) takes the value defined by \( d_i = \max\{0, t_i - \hat{t}_i\} \). From constraints \( d_i \geq 0 \) and \( d_i \geq t_i - \hat{t}_i \), we obtain \( d_i \geq \max\{0, t_i - \hat{t}_i\} \). Moreover, the coefficients \( g_i \) in the objective function are non negative because they represent the passengers and taking into account that it is a minimization problem, the solution will be reached in minimum values of \( d_i \). Therefore, \( d_i = \max\{0, t_i - \hat{t}_i\} \) holds.

The disjunctive constraints are modelled using the binary variables \( y_{ij} \).

Figure 3.2 illustrates the proof. Let \( \{t_i\}_{i \in N} \) the operational times of a feasible solution. Then, for each pair of operations \( h \) and \( k \) holds

\[ |t_k - t_h| \leq t_n - t_0 \] (3.6)

Assume that \( y_{ij} = 1 \) for an alternative arc \(((i, j), (h, k)) \in A\). In this case the constraints associated with this pair of alternative arcs transform into

\[ \begin{align*}
& \quad t_j - t_i \geq f_{ij} \quad \text{ (3.7)} \\
& \quad t_k - t_h \geq f_{hk} - M \quad \text{ (3.8)}
\end{align*} \]

Now it will be proved that Equation (3.8) is always satisfied i.e. the constraint is redundant. In a feasible solution, various jobs occur simultaneously, so the time necessary for carrying out all the operations, \( t_n - t_0 \), would be less than a sequential execution of the complete schedule. That is:

\[ t_n - t_0 \leq \sum_{(i, j) \in F} f_{ij} + \sum_{((i, j), (h, k)) \in A} (f_{ij} + f_{hk}) \] (3.9)
Constraints (6.3) and (3.9) imply:

\[- \sum_{(i,j) \in F} f_{ij} - \sum_{((i,j),(h,k)) \in A} (f_{ij} + f_{hk}) \leq t_k - t_h \leq \sum_{(i,j) \in F} f_{ij} + \sum_{((i,j),(h,k)) \in A} (f_{ij} + f_{hk}) \leq t_k - t_h \]

(3.10)

Constraint (3.8) is guaranteed due to the choice of $M$ which is a sufficiently large constant similar to the used in Samà et al. (2013):

\[f_{hk} - M = f_{hk} - \sum_{(i,j) \in F} f_{ij} - 2 \sum_{((i,j),(h,k)) \in A} (f_{ij} + f_{hk}) \leq - \sum_{(i,j) \in F} f_{ij} + \sum_{((i,j),(h,k)) \in A} (f_{ij} + f_{hk}) \leq t_k - t_h \]

(3.11)

In this case train $A$ is sequenced before $B$ in the track segment (see Figure 3.2). In the other case $y_{ij} = 0$ train $B$ will be sequenced before train $A$ because the other disjunctive constraint $t_k - t_h \geq f_{hk}$ is satisfied.
3.4 Heuristic algorithms

The formulation shown in the previous section using integer linear programming is useful for applying exact algorithms like Branch-and-Bound or Branch-and-Cut. However, the problem addressed in this Chapter is a particular case of **job-shop problem with limited buffer capacities** called **blocking job-shop problem** in which all buffers have capacity 0. Papadimitriou and Kanellakis (1980) showed that even the two-machine flow-shop problem with a limited buffer between the first and the second machine is strongly \( NP \)-hard. This fact suggests solving the blocking job-shop problem with heuristics in situations in which computational time is a key factor.

The literature contains several heuristics which could be adapted and used for the railway conflict resolution problem. Two popular approaches are: **First Come First Served (FCFS)** and **Avoid Most Critical Completion Time (AMCC)** [Mascis and Pacciarelli (2002), D’Ariano (2008)].

In FCFS a train has access to a resource in strict accordance with its arrival time, and the other trains which wish to enter a block section must wait until it is released. The traditional way in which a block section is given to a train is by strict order of arrival. This method has the advantage of simplicity, but it can lead to major disadvantages in the form of delays. For example, if a slow train reaches a block section first and its operation time is 10 minutes, this will cause a serious delay to a later train which arrives at the same block section a minute later and has an operation time of 5 minutes. This method does not take note of any delays in the rescheduling process, but an effect of this neglect is that delays can spread easily. Another problem with this principle is that it more easily leads to deadlock.

FCFS is, however, useful for detecting conflicts, since it corresponds to normal working, without the involvement of the dispatcher. This algorithm is used by the SIMEIFER tool presented in Chapter 2 to simulate the working of the system and to detect conflicts.

AMCC is a greedy algorithm based on global system information which tries to find a feasible solution to the scheduling problem. Firstly the finishing times for each route are studied for each pair of alternative arcs on the graph. Once the times are calculated, the higher time is identified and its use is forbidden, and the other arc is chosen.

### 3.4.1 Avoid Most Delayed Alternative Arc (AMDAA)

In this section, AMCC is adapted to the alternative graph model presented in Section 3.3. This heuristic is called **Avoid Most Delayed Alternative Arc (AMDAA)**. AMDAA attempts to reduce the total passenger delay at the destination station.
A selection $S$ is a set of arcs obtained from $A$ by choosing at most one arc per alternative pair. The selection is called complete if only one arc of the pair is chosen. Given a pair of arcs $a = (u, v) \in A$, $u$ is chosen in $S$ if $u \in S$, and $v$ is forbidden in $S$ if $v \in S$. If neither of the two arcs is chosen, the pair is called unselected pair. The set of unselected pairs is denoted as $A'$. For each alternative arc $w^*$, we therefore associate a list of implied alternative arcs $Stat(w^*)$.

Two static implication rules are defined to add an arc to the set $Stat(w^*)$. Let $B_1$ and $B_2$ be the block sections associated with pairs $((a, b), (c, d))$ and $((i, j), (h, k))$ respectively, and let $T_1$ and $T_2$ be two trains. $T_1$ executes $b$ before $i$, it means that nodes $b$ and $i$ are associated with train $T_1$ and are connected by a directed path of fixed arcs. $T_2$ executes $j$ before $a$, which means that nodes $j$ and $a$ are associated with train $T_2$ and are connected by a directed path of fixed arcs. $T_1$ and $T_2$ pass through the two block sections that $((a, b), (c, d))$ and $((i, j), (h, k))$ refer to.

The two rules that define a static implication and the addition of an arc to the set $Stat(w^*)$, developed by D’Ariano (2008) are:

1. *(Trains travelling in the same direction).* If $B_1$ and $B_2$ are adjacent block sections, traversed by $T_1$ and $T_2$ in the same order, then nodes $a$ and $j$, and $c$ and $k$, respectively, coincide. Thus, $(a, b) \in Stat((h, k))$, $(h, k) \in Stat((a, b))$, $(c, d) \in Stat((i, j))$ and $(i, j) \in Stat((c, d))$.

2. *(Trains travelling in opposite direction).* If $T_1$ and $T_2$ pass both through $B_1$ and $B_2$ in opposite directions then, $(h, k) \in Stat((a, b))$ and $(c, d) \in Stat((i, j))$.

Given a selection $S$, $G(S)$ shows the graph $(N, F \cup S)$. The selection will be consistent if in graph $G(S)$ there are no cycles, that is, a number of adjacent arcs where the same arc is not traversed twice, and which returns to the starting point. A solution is defined as a consistent selection, that is, a selection which has taken one arc from each pair of arcs present in $A$ and there are no cycles.

The selection $S$ is augmented in two ways. The first is by choosing a pair of unselected alternative arcs $a = (u, v)$ such that the total delays produced by this pair on passengers in the system is maximum, selecting the arc $w^*$ that minimizes it. The second is that all the static implications on the other alternative arcs contained in a given set $Stat(w^*)$ are added to $S$. Each time the selection $S$ is augmented, the start time of the operations related to the alternative arcs added to $S$ is calculated.
3.4. Heuristic algorithms

More formally, let \( a = (u, v) \in A' \), and let \( S \) be the current selection, then

\[
Z_u = \sum_{i \in N(S)} g_i \max \{0, t_i^u - \hat{t}_i\} \\
Z_v = \sum_{i \in N(S)} g_i \max \{0, t_i^v - \hat{t}_i\} \\
Z_a = \max \{Z_u, Z_v\}
\]

(3.12)

(3.13)

(3.14)

where \( t_i^u, t_i^v \) are the start time of the operation \( i \) on \( G(S \cup \{u\}) \) and \( G(S \cup \{v\}) \) respectively. AMDAA chooses the unselected pair as:

\[
a^* = \text{Arg maximize} \ Z_a \quad a = (u, v) \in A'
\]

(3.15)

Thus the new selection is

\[ S = S \cup \{w^*\} \]

where

\[
a^* = (u^*, v^*) \text{ and } Z_{w^*} = \text{Minimize} \{Z_{u^*}, Z_{v^*}\}. \]

(3.16)

The main difference between the AMCC and AMDAA algorithms is how they choose the arcs that will be selected. While in each iteration the AMCC algorithm enhances selection \( S \) by choosing a pair of unselected alternative arcs so as to minimize the makespan value, the AMDAA algorithm chooses alternative arcs which minimize the delay caused to passengers.

Potential deadlocks are handled in the AMCC and AMDAA algorithms the same way, defining the existence of a positive length cycle in selection \( S \) as a deadlock situation and modifying the alternative arc selected which this cycle produces. Given the nodes \( i, j \in N(F) \), \( l^S(i, j) \) denotes the length of the longest path from \( i \) to \( j \) on \( G(S) \). Consider a selection \( S \) and two unselected alternative pairs \( ((a, b), (c, d)) \) and \( ((i, j), (h, k)) \). If \( f_{ab} + l^S(b, i) + f_{ij} + l^S(j, a) \geq 0 \), then arc \( (h, k) \) is implied by selection \( S \cup \{(a, b)\} \) and arc \( (c, d) \) is implied by selection \( S \cup \{(i, j)\} \). Consequently the selection \( S \cup \{(a, b), (i, j)\} \) contains a positive length cycle.

The complete AMDAA pseudocode can be seen in Algorithm 2 which computes one feasible solution. This algorithm needs at first a pre-generation process in which the alternative graph \( G \) is created using the predefined routes of each train as the aspect which represents the directed arcs \( F \) between nodes \( N \) and the alternative arcs \( A \), and calculating all the values of these arcs considering the length of track segments and the speed of the trains.
CHAPTER 3. A DEMAND-BASED WEIGHTED TRAIN DELAY APPROACH FOR RESCHEDULING RAILWAY NETWORKS IN REAL TIME

Algorithm 2 AMDAA algorithm

Input: An alternative graph \( G = (N, F, A) \). Let \( S = \{\emptyset\} \) be the initial selection, and let \( A' = A \), \( cycle = false \).

1: \textbf{while} \( A' \neq \emptyset \) \textbf{and} \( cycle = false \) \textbf{do}

2: \hspace{1em} Let \( a^* \) be the alternative defined in Eq. (3.15).

3: \hspace{1em} Choose the arc \( w^* \) defined in Eq. (3.16).

4: \hspace{1em} Let \( S' := S \cup \{w^*\}, A' := A \setminus \{a^*\} \).

5: \hspace{1em} Select all arcs in set \( Stat(w^*) \).

6: \hspace{1em} if there is a cycle in \( G(S') \) or an arc from \( Stat(w^*) \) is forbidden then

7: \hspace{2em} Let \( S' := S \cup \{\tilde{w}^*\} \), where \( a^* = (w^*, \tilde{w}^*) \).

8: \hspace{2em} Select all arcs in the set \( Stat(\tilde{w}^*) \).

9: \hspace{2em} if there is a cycle in \( G(S') \) or an arc from \( Stat(\tilde{w}^*) \) is forbidden then

10: \hspace{3em} \( cycle = true \) (The procedure failed to compute a feasible solution).

11: \hspace{2em} end if

12: \hspace{1em} \( S := S' \).

13: \hspace{1em} end if

14: \textbf{end while}

15: \textbf{return} A consistent selection \( S \) or infeasible solution

3.4.2 Illustrative example

This section shows an example that will help in illustrating the difference between AMCC and AMDAA algorithms.

Consider first the network shown in Figure 3.3. This network has 5 stations and 2 trains which coincide in track segment 3 and their travel time in each block section is 105s and 100s respectively. Also the demand for trips \( S_1 \rightarrow S_3, S_1 \rightarrow S_4, S_2 \rightarrow S_3, S_2 \rightarrow S_5 \) is known. The demand is \( G_1 \) for the origin-destination pairs \( S_1 \rightarrow S_3, S_1 \rightarrow S_4 \), and \( G_2 \) for the pairs \( S_2 \rightarrow S_3, S_2 \rightarrow S_5 \).

The feasible set comprises only two solutions. One is that train 1 has precedence over train 2 in station 3, so alternative arc \((4, 9)\) is selected. This selection produces a delay of 115 seconds in train 2 as the planned time to reach the station is 200 but it must wait until train 1 leaves the station at instant 315. The other solution is that train 2 has precedence over train 1 in station 3, i.e. the alternative arc \((10, 3)\) is chosen. This selection produces a delay of 90 seconds in train 1 as the planned time to reach the station is 210 but it must wait until train 2 leaves the station at instant 300. Both heuristics explore the feasible set selecting the solution which minimizes its objective function.

The AMCC algorithm evaluates the solutions without considering demand and the results are 115s for \((4, 9)\) selection and 90s for \((10, 3)\) selection, so this algorithm selects alternative arc \((10, 3)\).

In the case of AMDAA algorithm the alternative arc selected will depend on the demand values. If the selected alternative arc is \((4, 9)\) then train 1
3.4. Heuristic algorithms

will run ahead of train 2 and it will cause a delay of 115s in each station to train 2, as the number of passengers who leave in each station is $G_2$ then the passengers will suffer $115 \times 2 \times G_2 = 230G_2$ seconds of delay, 115 * $G_2$ for each of the two destination stations. Otherwise, the selected arc would be (10,3) and train 2 will run ahead of train 1 and it will cause a delay of 90s in each station of train 1, as the number of passengers who leave in each station is $G_1$ then its total delay will be $90 \times 2 \times G_1 = 180G_1$ seconds, 90 * $G_1$ for each of the two destination stations. Mathematically,

$$Z_u = 230G_2$$

$$Z_v = 180G_1$$

where $u = (4,9)$ and $v = (10,3)$. Therefore the solution selected by AMDAA between the two alternative arcs will depend on the values of the demand parameters $G_1$ and $G_2$. If $230G_2 < 180G_1$ arc (4,9) will be selected. Otherwise, arc (10,3) will be selected. Figure 3.4 depicts the solution found by AMDAA depending on the parameters $G_1$ and $G_2$. The black line represent the solutions of the equation $230G_2 = 180G_1$ dividing the complete set of feasible solutions depending on the values given to $G_1$ and $G_2$ for AMDAA algorithm. The grey part presents the infinite set of solutions in which the AMDAA algorithm will obtain the same solution as the AMCC algorithm, and the white part represents the infinite set in which they differ. It is worth noting that the case when $G_1 = G_2$ represents a situation in which only the delays of all trains at stations are taken into account. This case belongs to the grey part of the picture.
CHAPTER 3. A DEMAND-BASED WEIGHTED TRAIN DELAY APPROACH FOR RESCHEDULING RAILWAY NETWORKS IN REAL TIME

3.5 Computational experiments

The network used for the computational experiments is the Madrid local rail network, called Renfe Cercanías Madrid. It makes up 60 percent of the local networks in Spain, 40 percent of rail traffic and is able to move more than a million passengers per day. One of the most important actions which have been undertaken recently consisted of the building of new lines to Barajas airport from the station of Chamartín and three stations, Manoteras, Valdebebas and Barajas T-4 on the route.

This part of the network, routes followed by trains and conflict zones are showed in Figure 3.5. This infrastructure comprises 93 block sections and is used by 475 trains in an operational period of 20 hours, which means an average of 24 trains running on the infrastructure per hour, with maxima of 36 and minima of 3 trains per hour.

Because the relevant interval for a rail dispatcher for real-time purposes in regional railway systems, because of their journey time, is estimated
3.5. Computational experiments

at something less than an hour, the numerical comparison will study the following two time windows, firstly the time horizon is divided into 20 intervals of 1 hour and solved individually, next the time horizon is divided into 10 intervals of 2 hours and solved individually. It is a rolling time horizon approach, and in this case the connection between time interval and number of trains is related to the predefined schedule, showing the number of trains that will start their journey in each interval. The case study has been obtained from the perturbation of the original timetable. A set of random events is applied to 65 percent of trains causing some of the trains to make their journey at a slower speed (5 to 30 percent reduction) than the speed planned for them. The data concerning the network and the complete disturbance scenario can be downloaded from the links in Appendix B.

To carry out the computational experiments the SOFGA tool\(^1\) ("Secuenciacion de Operaciones Ferroviarias mediante Grafos Alternativos" which means "Railway Operations Sequencing with Alternative Graphs") was developed to implement the models and algorithms mentioned in previous sections. The tool has been codified using MATLAB. Figure 3.6 shows a capture of the window of SOFGA tool. Area 1 represents the part the user uses for loading the XML files which contain the data of the initial schedule. Area 2 is used for selecting the intervals in which the user wants to work. Area 3 is used for applying the different algorithms to the selected interval and view the obtained solution.

![Figure 3.6: The SOFGA tool](image)

The FCFS, AMCC and AMDAA algorithms have been implemented with the SOFGA tool, and run on an Intel Core i3 2.40GHz with 4 GB of RAM, and CPLEX implemented in GAMS with a standard configuration and without a time limit for the computation for the integer linear programming models on an Intel i7 3.06 GHz with 16 GB of RAM. CPLEX uses a branch-and-cut

\(^1\)A video showing the working of SOFGA tool may be seen at Appendix B
algorithm for integer linear programming. In these experiments $BC_{MAKESPAN}$ and $BC_{DELAY}$ denote the branch-and-cut algorithm applied respectively to objective functions $Z_{MAKESPAN}$ and $Z_{DELAY}$.

The main goals of this computational study are to answer the following two questions:

1. What are the strengths and weaknesses of each algorithm for minimizing both passenger delay and makespan?
2. Why is the performance the way it is for each algorithm and how are the scenarios affecting it?

The answer to the first question is addressed in Experiment 1 and to the second in Experiment 2.

### 3.5.1 Experiment 1

This experiment is divided into two parts. The first test will be aimed at train delay minimization so the values of all the demands in the system have been set to $g_i = 1$. This value means that all trains have the same priority and all stations have the same relevance. This experiment focuses on comparing the performance of the algorithms in a schedule in which the demands are not known. The second test is focused on proving the capacity of the new model and AMDAA on a schedule with a known demand. Due to the fact that there is no available real data of the origin-destination demand matrix for the passengers, the demand variable $g_i$ has been defined for each train as a random number generated from 0 to 1. The file with the generated data has been uploaded, and can be downloaded from the link of Appendix B.

Results can be seen in Tables 3.3 and 3.4 for intervals of 1 hour, and in Table 3.5 for intervals of 2 hours. These tables show the number of trains which make their services in each time interval and the computational cost, which is the average time taken to compute the optimal solution in seconds (CPU) taking into account that the AMDAA and $BC_{DELAY}$ are launched twice, once per experiment, and the other algorithms only once, due to their insensitivity to demand values. Furthermore the values of the objective functions $Z_{MAKESPAN}$ and $Z_{DELAY}$, evaluated in seconds, for each algorithm and for each set of values of $g_i$. In order to clarify these results, Tables 3.1 and 3.2 show the mean relative error MRE:

$$MRE = \frac{1}{n} \sum_{i=1}^{n} \left( \frac{Z^i - Z^i_{BC}}{Z^i_{BC}} \right)$$  \hspace{1cm} (3.17)

for each algorithm with respect to the optimal value obtained. The parameter $n$ is the number of intervals evaluated, $Z^i$ the objective function value found
3.5. Computational experiments

by the algorithm tested in an interval $i$ and $Z_{BC}^i$, the objective function value found by the BC method in the interval $i$.

<table>
<thead>
<tr>
<th>$BC_{MAKESPAN}$</th>
<th>$BC_{DELAY}$</th>
<th>FCFS</th>
<th>AMCC</th>
<th>AMDAA</th>
</tr>
</thead>
<tbody>
<tr>
<td>$g_i=1$</td>
<td>rand $g_i$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$Z_{MAKESPAN}$</td>
<td>0</td>
<td>0.0519</td>
<td>1.5683</td>
<td>0.0435</td>
</tr>
<tr>
<td>$g_i=1$</td>
<td></td>
<td>0.0336</td>
<td>0.0336</td>
<td>1.2524</td>
</tr>
<tr>
<td>$Z_{DELAY}$</td>
<td>1.6031</td>
<td>0</td>
<td>-</td>
<td>0.0039</td>
</tr>
<tr>
<td>rand $g_i$</td>
<td></td>
<td>0.0311</td>
<td>0.0311</td>
<td>-</td>
</tr>
<tr>
<td>2.9694</td>
<td></td>
<td>0.5278</td>
<td>0.5350</td>
<td>0.1767</td>
</tr>
</tbody>
</table>

Table 3.1: Relative error summary of results for 1–hour interval

<table>
<thead>
<tr>
<th>$BC_{MAKESPAN}$</th>
<th>$BC_{DELAY}$</th>
<th>FCFS</th>
<th>AMCC</th>
<th>AMDAA</th>
</tr>
</thead>
<tbody>
<tr>
<td>$g_i=1$</td>
<td>rand $g_i$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$Z_{MAKESPAN}$</td>
<td>0</td>
<td>0.1030</td>
<td>1.8384</td>
<td>0.0877</td>
</tr>
<tr>
<td>$g_i=1$</td>
<td></td>
<td>0.0350</td>
<td>0.0350</td>
<td>0.8456</td>
</tr>
<tr>
<td>$Z_{DELAY}$</td>
<td>3.4496</td>
<td>0</td>
<td>-</td>
<td>0.0026</td>
</tr>
<tr>
<td>rand $g_i$</td>
<td></td>
<td>0.0163</td>
<td>0.0163</td>
<td>-</td>
</tr>
<tr>
<td>8.4051</td>
<td></td>
<td>0.4631</td>
<td>0.4682</td>
<td>0.1199</td>
</tr>
</tbody>
</table>

Table 3.2: Relative error summary of results for 2–hour interval

As can be seen in Tables 3.1 and 3.2, these results show significant differences in more crowded intervals. With regard to the exact methods when $g_i = 1$ it is shown how the $BC_{DELAY}$ optimizes also the objective function $Z_{MAKESPAN}$ in most cases. It doesn’t occur so often vice versa, showing the main deficiency of $BC_{MAKESPAN}$ when $g_i = 1$, focusing only on the makespan of the worst train of the schedule, so if a train is highly delayed and there is no possibility of facing that delay, the solution given could not take into account the disruptions occurring between other trains.

With respect to the results for the exacts methods when the demand $g_i$ is generated randomly, it can be seen that the solutions found by the two exact approaches are meaningfully different, finding solutions based only on the objective function, which are worse if compared with the opposite objective.

With regard to heuristics, in the case of $g_i = 1$, it can be shown that the FCFS performs relatively well for this problem because the railway network is simple and presents few overtaking points, and it finds near-optimal solutions for $Z_{DELAY}$ objective function, but worse results if compared to $Z_{MAKESPAN}$. There are a few locations where the train ordering decision may be critical in terms of delay propagation. In this case, AMCC and AMDAA obtain the same results for all the cases.

In the case of the generated random demand, similar conclusions to those obtained with the exact algorithms are found. The main difference is that the FCFS algorithm finds worse solutions for the $Z_{DELAY}$ objective function than AMDAA. Moreover, comparing AMCC and AMDAA, the results show how each algorithm focuses on the improvement of its own objective function without taking others into account, obtaining different solutions in most cases.
In conclusion, heuristic algorithms are sufficiently robust when considering both objective functions. It can be seen that AMDAA and AMCC obtain good solutions in this test problem. It is worth noting that AMCC is a little faster than AMDAA because the minimization of the average delay requires more intensive data structure updates compared to the minimization of the maximum delay. Exact methods can be applied to a real-time context, but if they are applied, it should be considered that minimizing $Z_{\text{MAKESPAN}}$ could imply poor results for $Z_{\text{DELAY}}$ and vice versa.

Tables 3.3, 3.4 and 3.5 show how the optimization problem of the $Z_{\text{MAKESPAN}}$ objective function could obtain multiple optima due to the fact that other algorithms are capable of finding different solutions, because a different $Z_{\text{DELAY}}$ value is obtained, but having the optimal $Z_{\text{MAKESPAN}}$ value. This fact could motivate a bi-objective approach using both objective functions.

Figure 3.7 depicts a set of histograms which represent the number of trains in the solution obtained with a given final delay. Each histogram is associated with an algorithm and test, and is grouped by intervals of 30 seconds. These histograms are used for analysing the behaviour of the algorithms qualitatively. The main conclusion obtained is that the $BC_{\text{MAKESPAN}}$ algorithm adds more trains with an individual delay to the schedule while the other algorithms present a schedule with fewer delayed trains. Particularly in some cases the $BC_{\text{DELAY}}$ and AMDAA algorithms allow few trains with a delay more than 180 seconds to minimize the overall $Z_{\text{DELAY}}$ objective function.

### 3.5.2 Experiment 2

Experiment 2 discusses the viability of using the heuristic and exact methods in real problems. For this purpose each heuristic and exact method has been tested with 9 intervals of 1 to 9 hours to evaluate, starting each of them at the initial hour of the schedule seen in Experiment 1, adding in each new experiment the trains of the next 1 hour interval. The results obtained can be seen in Table 3.6. Each row presents the number of intervals or hours evaluated, the number of trains and alternative arcs evaluated and the computational cost in seconds of each algorithm for this interval.

This result shows that exact methods are viable for computing medium-sized problems in a real-time horizon and that heuristics could obtain near-optimal results for large problems with a reduced computational cost.

Figure 3.8 shows graphically the evolution of computational cost for each algorithm with regard to the number of trains (or alternative arcs) evaluated. It can be seen that exact algorithms’ computational costs have exponential growth related to the number of alternative arcs of the interval, showing that the exact methods are viable for some sizes of the problem but for large sizes the computational cost may become prohibitive because of the number of operations required to find the exact solution, thus heuristic algorithms are needed.
3.6 Conclusions

This Chapter presents a novel weighted train delay based on demand approach for rescheduling railway systems. The model seeks to minimize the sum of the accumulated delays that passengers suffer on the railway network. This approach allows us, if an Origin-Destination matrix is available, to take demand into account in the rescheduling process.

In this work the AMCC algorithm, which was designed for the optimization of makespan, has been adapted to the AMDAA algorithm for minimizing the delays and so maximizing customer satisfaction.

A computational study with AMCC, FCFS, AMDAA and branch-and-cut methods has been carried out using a perturbation of the original timetable planned by Renfe Cercanias Madrid in order to test the performance of the algorithms for resolving railway clashes and to ascertain their goodness.

These trials have shown that if demand is unknown (i.e. $g_i = 1$) heuristic algorithms provide good solutions to both makespan and delay objectives, but are unable to reach the goodness of the solutions given by exact

Figure 3.7: Number of trains delayed per algorithm

3.6 Conclusions
### Table 3.3: Numerical results for 1–hour interval (1/2)

<table>
<thead>
<tr>
<th>Temporal Interval</th>
<th>Number of Trains</th>
<th>Algorithm</th>
<th>CPU (in seconds)</th>
<th>Exp. $q_i = 1$</th>
<th>Exp. rand $q_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>$Z_{\text{MAKESPAN}}$</td>
<td>$Z_{\text{DELAY}}$</td>
<td>$Z_{\text{MAKESPAN}}$</td>
</tr>
<tr>
<td>1</td>
<td>5</td>
<td>BC$_{\text{MKESPAN}}$</td>
<td>0.31</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AMCC</td>
<td>0.14</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AMDAA</td>
<td>0.22</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>26</td>
<td>BC$_{\text{MKESPAN}}$</td>
<td>0.92</td>
<td>27.82</td>
<td>55.64</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FCFS</td>
<td>0.86</td>
<td>27.82</td>
<td>55.64</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AMCC</td>
<td>0.64</td>
<td>27.82</td>
<td>55.64</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AMDAA</td>
<td>0.77</td>
<td>27.82</td>
<td>55.64</td>
</tr>
<tr>
<td>3</td>
<td>34</td>
<td>BC$_{\text{MKESPAN}}$</td>
<td>2.39</td>
<td>89.21</td>
<td>2168.76</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AMCC</td>
<td>0.14</td>
<td>89.91</td>
<td>774.56</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FCFS</td>
<td>1.06</td>
<td>120.21</td>
<td>846.28</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AMDAA</td>
<td>1.29</td>
<td>120.21</td>
<td>846.28</td>
</tr>
<tr>
<td>4</td>
<td>36</td>
<td>BC$_{\text{MKESPAN}}$</td>
<td>2.59</td>
<td>110.54</td>
<td>3180.53</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AMCC</td>
<td>1.72</td>
<td>168.09</td>
<td>661.63</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FCFS</td>
<td>0.15</td>
<td>134.4</td>
<td>685.23</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AMDAA</td>
<td>1.26</td>
<td>148.02</td>
<td>775.36</td>
</tr>
<tr>
<td>5</td>
<td>32</td>
<td>BC$_{\text{MKESPAN}}$</td>
<td>1.5</td>
<td>90.69</td>
<td>2827.17</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AMCC</td>
<td>0.14</td>
<td>89.91</td>
<td>774.56</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FCFS</td>
<td>0.92</td>
<td>92.09</td>
<td>586.12</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AMDAA</td>
<td>2.39</td>
<td>92.09</td>
<td>586.12</td>
</tr>
<tr>
<td>6</td>
<td>24</td>
<td>BC$_{\text{MKESPAN}}$</td>
<td>1.16</td>
<td>92.42</td>
<td>522.42</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AMCC</td>
<td>1.36</td>
<td>450.93</td>
<td>147.82</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FCFS</td>
<td>0.08</td>
<td>450.93</td>
<td>147.82</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AMDAA</td>
<td>0.51</td>
<td>450.93</td>
<td>147.82</td>
</tr>
<tr>
<td>7</td>
<td>21</td>
<td>BC$_{\text{MKESPAN}}$</td>
<td>0.66</td>
<td>28.38</td>
<td>85.14</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AMCC</td>
<td>0.77</td>
<td>28.38</td>
<td>85.14</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FCFS</td>
<td>0.07</td>
<td>28.38</td>
<td>85.14</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AMDAA</td>
<td>0.45</td>
<td>28.38</td>
<td>85.14</td>
</tr>
<tr>
<td>8</td>
<td>22</td>
<td>BC$_{\text{MKESPAN}}$</td>
<td>0.81</td>
<td>23.28</td>
<td>264.48</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AMCC</td>
<td>0.89</td>
<td>23.28</td>
<td>171.38</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FCFS</td>
<td>0.08</td>
<td>23.28</td>
<td>171.38</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AMDAA</td>
<td>0.41</td>
<td>23.28</td>
<td>171.38</td>
</tr>
<tr>
<td>9</td>
<td>20</td>
<td>BC$_{\text{MKESPAN}}$</td>
<td>0.5</td>
<td>23.28</td>
<td>171.38</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AMCC</td>
<td>0.5</td>
<td>23.28</td>
<td>171.38</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FCFS</td>
<td>0.06</td>
<td>23.28</td>
<td>171.38</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AMDAA</td>
<td>0.36</td>
<td>23.28</td>
<td>171.38</td>
</tr>
<tr>
<td>10</td>
<td>30</td>
<td>BC$_{\text{MKESPAN}}$</td>
<td>0.4</td>
<td>23.28</td>
<td>171.38</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AMCC</td>
<td>0.16</td>
<td>78</td>
<td>3316.68</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FCFS</td>
<td>1.03</td>
<td>78</td>
<td>432.05</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AMDAA</td>
<td>0.03</td>
<td>78</td>
<td>432.05</td>
</tr>
</tbody>
</table>

Moreover, an exact method for minimizing makespan can lead to unsatisfactory solutions from the point of view of customer satisfaction in most cases. In the case of randomly generated demand values, each algorithm or exact method only considers its main purpose, obtaining poor solutions for the rest, so for future work a multiobjective approach should be studied.

From the optimization standpoint, the heuristic and exact methods provide acceptable solutions within a reasonable time window for the temporal horizon of one hour, which is the time horizon of practical interest.
3.6. Conclusions

Table 3.4: Numerical results for 1–hour interval (2/2)

<table>
<thead>
<tr>
<th>Temporal Interval</th>
<th>Number of trains</th>
<th>Algorithm</th>
<th>CPU (in seconds)</th>
<th>Exp. $g_1 = 1$ MAKESPAN</th>
<th>Exp. $g_1 = 1$ DELAY</th>
<th>Exp. rand $g_2$ MAKESPAN</th>
<th>Exp. rand $g_2$ DELAY</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>26</td>
<td>BCMAKESPAN</td>
<td>1.66</td>
<td>79.64</td>
<td>692.87</td>
<td>79.64</td>
<td>214.86</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BCDELAY</td>
<td>0.67</td>
<td>79.64</td>
<td>226.21</td>
<td>79.64</td>
<td>142.45</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FCFS</td>
<td>0.09</td>
<td>79.64</td>
<td>226.21</td>
<td>79.64</td>
<td>177.99</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AMCC</td>
<td>0.58</td>
<td>79.64</td>
<td>226.21</td>
<td>90.69</td>
<td>142.45</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AMDAA</td>
<td>0.71</td>
<td>79.64</td>
<td>226.21</td>
<td>79.64</td>
<td>87.58</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BCMAKESPAN</td>
<td>1.33</td>
<td>27.82</td>
<td>248.65</td>
<td>27.82</td>
<td>87.58</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BCDELAY</td>
<td>0.75</td>
<td>27.82</td>
<td>161.66</td>
<td>148.38</td>
<td>60.02</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FCFS</td>
<td>0.1</td>
<td>27.82</td>
<td>161.66</td>
<td>27.82</td>
<td>68.04</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AMCC</td>
<td>0.64</td>
<td>27.82</td>
<td>181.66</td>
<td>79.64</td>
<td>73.48</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AMDAA</td>
<td>0.77</td>
<td>27.82</td>
<td>181.66</td>
<td>148.38</td>
<td>65.46</td>
</tr>
<tr>
<td>12</td>
<td>27</td>
<td>BCMAKESPAN</td>
<td>1.39</td>
<td>27.82</td>
<td>186.91</td>
<td>27.82</td>
<td>22.56</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BCDELAY</td>
<td>0.67</td>
<td>27.82</td>
<td>79.64</td>
<td>27.82</td>
<td>19.79</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FCFS</td>
<td>0.08</td>
<td>27.82</td>
<td>79.64</td>
<td>27.82</td>
<td>19.79</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AMCC</td>
<td>0.49</td>
<td>27.82</td>
<td>79.64</td>
<td>27.82</td>
<td>19.79</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AMDAA</td>
<td>0.6</td>
<td>27.82</td>
<td>79.64</td>
<td>27.82</td>
<td>19.79</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BCMAKESPAN</td>
<td>1.38</td>
<td>77.18</td>
<td>505.85</td>
<td>77.18</td>
<td>175.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BCDELAY</td>
<td>0.81</td>
<td>77.18</td>
<td>229.42</td>
<td>218.99</td>
<td>167.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FCFS</td>
<td>0.1</td>
<td>77.18</td>
<td>229.42</td>
<td>87.82</td>
<td>180.11</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AMCC</td>
<td>0.6</td>
<td>77.18</td>
<td>229.42</td>
<td>77.18</td>
<td>178.68</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AMDAA</td>
<td>0.77</td>
<td>77.18</td>
<td>229.42</td>
<td>77.18</td>
<td>178.68</td>
</tr>
<tr>
<td>13</td>
<td>24</td>
<td>BCMAKESPAN</td>
<td>1.19</td>
<td>32.42</td>
<td>234.78</td>
<td>32.42</td>
<td>186.54</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BCDELAY</td>
<td>0.83</td>
<td>32.42</td>
<td>190.52</td>
<td>140.27</td>
<td>106.44</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FCFS</td>
<td>0.11</td>
<td>32.42</td>
<td>190.52</td>
<td>32.42</td>
<td>144.44</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AMCC</td>
<td>0.74</td>
<td>32.42</td>
<td>190.52</td>
<td>32.42</td>
<td>144.44</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AMDAA</td>
<td>0.9</td>
<td>32.42</td>
<td>190.52</td>
<td>32.42</td>
<td>144.44</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BCMAKESPAN</td>
<td>1.34</td>
<td>32.42</td>
<td>253.3</td>
<td>32.42</td>
<td>105.66</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BCDELAY</td>
<td>0.8</td>
<td>32.42</td>
<td>243.3</td>
<td>32.42</td>
<td>80.74</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FCFS</td>
<td>0.1</td>
<td>32.42</td>
<td>243.3</td>
<td>32.42</td>
<td>80.74</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AMCC</td>
<td>0.64</td>
<td>32.42</td>
<td>263.3</td>
<td>32.42</td>
<td>89.52</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AMDAA</td>
<td>0.78</td>
<td>32.42</td>
<td>263.3</td>
<td>32.42</td>
<td>89.52</td>
</tr>
<tr>
<td>14</td>
<td>27</td>
<td>BCMAKESPAN</td>
<td>0.78</td>
<td>20.63</td>
<td>234.78</td>
<td>20.63</td>
<td>44.93</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BCDELAY</td>
<td>1.14</td>
<td>20.63</td>
<td>20.63</td>
<td>20.63</td>
<td>106.44</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FCFS</td>
<td>0.08</td>
<td>20.63</td>
<td>20.63</td>
<td>20.63</td>
<td>8.24</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AMCC</td>
<td>0.49</td>
<td>20.63</td>
<td>20.63</td>
<td>20.63</td>
<td>8.24</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AMDAA</td>
<td>0.6</td>
<td>20.63</td>
<td>20.63</td>
<td>20.63</td>
<td>16.66</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BCMAKESPAN</td>
<td>0.56</td>
<td>23.28</td>
<td>147.82</td>
<td>23.28</td>
<td>54.59</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BCDELAY</td>
<td>0.77</td>
<td>23.28</td>
<td>46.55</td>
<td>147.82</td>
<td>36.14</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FCFS</td>
<td>0.07</td>
<td>23.28</td>
<td>46.55</td>
<td>23.28</td>
<td>43.86</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AMCC</td>
<td>0.33</td>
<td>23.28</td>
<td>46.55</td>
<td>23.28</td>
<td>43.86</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AMDAA</td>
<td>0.4</td>
<td>23.28</td>
<td>46.55</td>
<td>23.28</td>
<td>43.86</td>
</tr>
<tr>
<td>15</td>
<td>29</td>
<td>BCMAKESPAN</td>
<td>0.7</td>
<td>20.63</td>
<td>61.9</td>
<td>20.63</td>
<td>44.93</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BCDELAY</td>
<td>0.45</td>
<td>20.63</td>
<td>26.14</td>
<td>20.63</td>
<td>14.57</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FCFS</td>
<td>0.06</td>
<td>20.63</td>
<td>26.14</td>
<td>20.63</td>
<td>14.57</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AMCC</td>
<td>0.29</td>
<td>20.63</td>
<td>26.14</td>
<td>20.63</td>
<td>14.57</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AMDAA</td>
<td>0.35</td>
<td>20.63</td>
<td>26.14</td>
<td>20.63</td>
<td>14.57</td>
</tr>
<tr>
<td>16</td>
<td>27</td>
<td>BCMAKESPAN</td>
<td>0.31</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BCDELAY</td>
<td>0.53</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FCFS</td>
<td>0.01</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AMCC</td>
<td>0.02</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AMDAA</td>
<td>0.02</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

to rail traffic controllers for real-time purposes in regional railway systems because of their journey time. It has also been proved that with exact methods the computational cost can grow very quickly, making them impractical, and requiring heuristic algorithms such as those proposed in this Chapter.
### Table 3.5: Numerical results for 2–hours interval

<table>
<thead>
<tr>
<th>Temporal Interval</th>
<th>Number of trains</th>
<th>Algorithm</th>
<th>CPU (in seconds)</th>
<th>Exp. $q_t = 1$</th>
<th>Exp. rand $q_t$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>31</td>
<td>BC$_{DELSAP}$</td>
<td>1.3</td>
<td>27.82</td>
<td>681.96</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BC$_{DELAY}$</td>
<td>1.03</td>
<td>27.82</td>
<td>55.64</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FCFS</td>
<td>0.14</td>
<td>27.82</td>
<td>55.64</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AMCC</td>
<td>0.97</td>
<td>27.82</td>
<td>55.64</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AMDA</td>
<td>1.25</td>
<td>27.82</td>
<td>55.64</td>
</tr>
<tr>
<td></td>
<td>70</td>
<td>BC$_{DELSAP}$</td>
<td>13.67</td>
<td>110.54</td>
<td>8697.22</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BC$_{DELAY}$</td>
<td>36.44</td>
<td>168.09</td>
<td>1436.19</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FCFS</td>
<td>0.53</td>
<td>134.39</td>
<td>1459.79</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AMCC</td>
<td>5.79</td>
<td>148.02</td>
<td>1649.65</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AMDA</td>
<td>6.91</td>
<td>148.02</td>
<td>1649.66</td>
</tr>
<tr>
<td></td>
<td>56</td>
<td>BC$_{DELSAP}$</td>
<td>6.09</td>
<td>92.42</td>
<td>6131.14</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BC$_{DELAY}$</td>
<td>5.45</td>
<td>139.64</td>
<td>959.84</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FCFS</td>
<td>0.32</td>
<td>139.64</td>
<td>959.84</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AMCC</td>
<td>3.9</td>
<td>92.42</td>
<td>1017.05</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AMDA</td>
<td>4.59</td>
<td>92.42</td>
<td>1017.05</td>
</tr>
<tr>
<td></td>
<td>43</td>
<td>BC$_{DELSAP}$</td>
<td>2.39</td>
<td>28.38</td>
<td>859.47</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BC$_{DELAY}$</td>
<td>1.78</td>
<td>28.38</td>
<td>228.14</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FCFS</td>
<td>0.2</td>
<td>28.38</td>
<td>228.14</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AMCC</td>
<td>1.66</td>
<td>28.38</td>
<td>228.14</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AMDA</td>
<td>2.05</td>
<td>28.38</td>
<td>228.14</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>BC$_{DELSAP}$</td>
<td>3.42</td>
<td>78</td>
<td>4036.61</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BC$_{DELAY}$</td>
<td>2.8</td>
<td>78</td>
<td>460.63</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FCFS</td>
<td>0.26</td>
<td>78</td>
<td>460.43</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AMCC</td>
<td>2.43</td>
<td>78</td>
<td>460.43</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AMDA</td>
<td>2.95</td>
<td>78</td>
<td>460.43</td>
</tr>
<tr>
<td></td>
<td>53</td>
<td>BC$_{DELSAP}$</td>
<td>3.9</td>
<td>79.64</td>
<td>4128.53</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BC$_{DELAY}$</td>
<td>3.86</td>
<td>79.64</td>
<td>387.87</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FCFS</td>
<td>0.29</td>
<td>79.64</td>
<td>387.87</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AMCC</td>
<td>2.77</td>
<td>79.64</td>
<td>407.87</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AMDA</td>
<td>3.92</td>
<td>79.64</td>
<td>407.87</td>
</tr>
<tr>
<td></td>
<td>51</td>
<td>BC$_{DELSAP}$</td>
<td>3.45</td>
<td>77.18</td>
<td>4339.31</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BC$_{DELAY}$</td>
<td>3.24</td>
<td>77.18</td>
<td>267.23</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FCFS</td>
<td>0.27</td>
<td>77.18</td>
<td>267.23</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AMCC</td>
<td>2.52</td>
<td>77.18</td>
<td>267.23</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AMDA</td>
<td>3.08</td>
<td>77.18</td>
<td>267.23</td>
</tr>
<tr>
<td></td>
<td>56</td>
<td>BC$_{DELSAP}$</td>
<td>4.03</td>
<td>32.42</td>
<td>1725.23</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BC$_{DELAY}$</td>
<td>4.13</td>
<td>32.42</td>
<td>433.82</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FCFS</td>
<td>0.32</td>
<td>32.42</td>
<td>433.82</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AMCC</td>
<td>3.17</td>
<td>32.42</td>
<td>453.82</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AMDA</td>
<td>3.86</td>
<td>32.42</td>
<td>453.82</td>
</tr>
<tr>
<td></td>
<td>44</td>
<td>BC$_{DELSAP}$</td>
<td>1.86</td>
<td>23.28</td>
<td>629.45</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BC$_{DELAY}$</td>
<td>1.9</td>
<td>23.28</td>
<td>67.19</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FCFS</td>
<td>0.21</td>
<td>23.28</td>
<td>67.19</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AMCC</td>
<td>1.74</td>
<td>23.28</td>
<td>67.19</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AMDA</td>
<td>2.16</td>
<td>23.28</td>
<td>67.19</td>
</tr>
<tr>
<td></td>
<td>21</td>
<td>BC$_{DELSAP}$</td>
<td>0.73</td>
<td>20.27</td>
<td>40.54</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BC$_{DELAY}$</td>
<td>0.56</td>
<td>20.27</td>
<td>26.14</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FCFS</td>
<td>0.07</td>
<td>20.27</td>
<td>26.14</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AMCC</td>
<td>0.4</td>
<td>20.27</td>
<td>26.14</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AMDA</td>
<td>0.49</td>
<td>20.27</td>
<td>26.14</td>
</tr>
</tbody>
</table>

### Table 3.6: Computational cost versus number of trains and amplitude of the time window
3.6. Conclusions

Figure 3.8: Computational cost comparison
Part II: Off-line approach
4. A framework for derivative free algorithm hybridization

PART I: ON-LINE

PART II: OFF-LINE

1. Introduction
2. Conflict detection
3. Conflict resolution
4. Hybrid framework
5. CMNL model
6. HSR-TTP model
7. Conclusions
4
A framework for derivative free algorithm hybridization

Column generation is a basic tool for the solution of large-scale mathematical programming problems. We present a class of column generation algorithms in which the columns are generated by derivative free algorithms, like population-based algorithms. This class can be viewed as a framework to define hybridization of free derivative algorithms. This framework has been illustrated in this Chapter using the Simulated Annealing (SA) and Particle Swarm Optimization (PSO) algorithms, combining them with the Nelder-Mead (NM) method. Finally a set of computational experiments has been carried out to illustrate the potential of this framework.

4.1 Introduction

Derivative free algorithms is a key area with an increasing interest due to its versatility and broad use in optimization problems. The main motivation for this Chapter is to propose a means to accelerate the convergence of these algorithms in order to be able to apply them to large-scale problems in a reasonable computational time.

Column generation (CG) García et al. (2003) is a classic means to attack the following (large-scale) constrained optimization problem:

\[
\minimize_{x \in X} f(x), \quad P(f, X)
\]

where the feasible region \( X \subseteq \mathbb{R}^N \) is non-empty and closed, and \( f : X \mapsto \mathbb{R} \) is a continuous function on \( X \). These algorithms follow two main steps:
1. Column Generation Problem (CGP). A relaxation of the original problem is constructed, based on current estimates $\tilde{x}$ of the optimal solution, which provides a bound to the optimal value of the original problem, a new column and information that indicates if the column should be introduced into the solution or if the process should stop. This problem can be defined by $P(\hat{f}, X)$ where $\hat{f}(x)$ is an approximation of $f(x)$ on $\tilde{x}$.

2. Restricted Master Problem (RMP). Previously generated columns define an inner approximation $\hat{X}$ of the feasible region $X$ and the new column $\tilde{y}$ is used to expand it. The original problem is approximated by $P(f, \hat{X})$ and its solution $\tilde{x}$ is used to define a new CG sub-problem which iterates the process.

Classic examples of this type of algorithm used for linearly and nonlinearly constrained problems are simplicial decomposition (SD) (see Von Hohenbalken (1977) and Marín (1995)) and Restricted Simplicial Decomposition (Ventura and Hearn (1993)). In these algorithms the relaxation is a linearization of the objective function while the feasible set is defined by the convex hull of the retained columns in the RMP. In this algorithm column generation is not based on pricing or dual information and is associated with multidimensional extensions of primal descent algorithms in NLP.

In García et al. (2003) a CG algorithm is proposed based on closed descent algorithms. In this framework the CGP is defined by a given algorithm and the RMP as a phase which tries to accelerate the algorithm used in CGP. RMP is solved by an algorithm capable of using the advantages of RMP to its favour, like a reduced number of variables and a simple set of restrictions.

In this algorithm the SD is obtained as a column running only one iteration of the Frank-Wolfe method Frank and Wolfe (1956). A key theoretical result is that the local rate of convergence of SD is governed by the local convergence rate of the method chosen for the solution of the RMP; thus a superlinear or quadratic convergence rate may be attained if a (projected) Newton method is used Hearn et al. (1987). Note that the rate of convergence of the Frank-Wolfe algorithm is sublinear.

InGarca-Rdenas et al. (2011) was carried out an experimental study focused on their computational efficiency of CG methods. Two types of test problems were considered, the first one is the nonlinear, capacitated single-commodity network flow problem, and the second one is a combined traffic assignment model. This paper validates this methodology in order to improve the performance of feasible direction and simplicial decomposition methods used in equilibrium assignment models.

The main contribution of this Chapter is to extend this framework to derivative free optimization methods for general optimization problems. To verify the goodness of the framework a set of computational tests have been performed with the algorithms: i) Nelder-Mead simplex method (NM)
4.2 The conceptual framework of hybridization

Nelder and Mead (1965) ii) Bohachevsky et al. Simulated Annealing (SA) Bohachevsky et al. (1986), and iii) Particle Swarm Optimization (PSO) Kennedy and Eberhart (1995), showing that the convergence rate and effectiveness of the heuristics algorithms can be improved by hybridization.

The Chapter is organized as follows. In Section 4.2 the proposed framework for derivative free optimization methods is explained, in Section 4.3 we discuss some instances of the hybrid CG algorithm, in Section 4.4 several computational experiments are reported, and finally in Section 4.5 we conclude with a discussion of our findings and future work.

4.2 The conceptual framework of hybridization

García et al. (2003) dealt with a class of CG algorithms in which the approximated objective function $\hat{f}(y)$ coincides with the original $f(y)$. This point of view leads to a CG algorithm which can be defined according to the following three key items: i) the algorithm to solve the $P(f,X)$ (denoted by $A_c$), ii) the algorithm applied to RMP (denoted by $A_r$) and iii) the means of stating the inner approximation $\hat{X}$. The algorithms $A_c$ and $A_r$ analysed in García et al. (2003) are descent primal methods and the $P(f,X)$ is a differentiable optimization problem. In this Chapter we extend this framework to derivative free optimization methods and for general optimization problems $P(f,X)$. We begin by stating the characteristics of the optimization algorithms $A_c$ and $A_r$ that can be used to solve the CGP($f,X$) or RMP($f,\hat{X}$).

A type of derivative free algorithm which works with populations generates an evolution of the population instead of generating a sequence of solutions (particles in Particle Swarm Optimization, an atom in Simulated Annealing, chromosomes in Genetic Algorithms, ants in Ant Colony Optimization, etc. Kennedy et al. (2001)). For this reason we consider algorithms based on populations, and if the cardinality of this population is one the classical optimization methods appear.

**Assumption 1 (Optimization Algorithm).** Let $P(f,Z)$ be an optimization problem and let $A$ be an optimization algorithm. This algorithm is defined as an iterative procedure which will be assumed to fulfill the following two conditions.

i) *(Feasible population).* This algorithm works on a population of particles $\tilde{Z} = \{z_1, \cdots, z_m\}$ which is modified iteratively. This algorithm is described by means of a point-to-set algorithmic mapping

\[
A : Z^m \mapsto 2^Z^m
\]

\[
\tilde{Z} \mapsto A(\tilde{Z})
\]  

(4.1)
where \( 2^{Z^m} \) is the power set of \( Z^m \). Also we denote

\[
A'(\tilde{Z}) := [A \circ \cdots \circ A](\tilde{Z})
\]

(4.2)

and the realization of \( t' \)-iterations generate a sequence of populations \( \tilde{Z}^1, \ldots, \tilde{Z}^{t'} \) such as \( \tilde{Z}^t := \{z_1^t, \ldots, z_m^t\} \in A(Z^{t-1}) \) of feasible points for any initialization \( \tilde{Z} \in Z^m \).

ii) (Convergent property). This algorithm is convergent for every \( \tilde{Z} \in Z^m \) in the following sense. Let \( Z^t \in A(t, Z) \), \( t = 1, 2, \ldots \) be the sequence of populations generated by \( A \) and let

\[
z^t := \text{Arg minimize} f(z)
\]

(4.3)

then

\[
d_{\text{SOL}(f, Z)}(z^t) := \left\{ \min_{z^* \in \text{SOL}(f, Z)} \|z^t - z^*\| \right\} \to 0
\]

(4.4)

where \( \text{SOL}(f, Z) \) denotes the set of global minimizers of \( P(f, Z) \).

The methods used in García et al. (2003) are descent closed algorithms which are convergent for differentiable convex programs. In this Chapter we generalize them to convergent algorithms for a given optimization problem. Properties i) and ii) guarantee the convergence of the algorithm in a finite number of iterations or a finite number of descents in the objective function. In the CG method the number of these descents is used to decide a change between the algorithms used. The following definition explains this concept.

**Definición 4.2.1** \((n\text{--}descent iteration)\). We say that an algorithm \( A \) achieves an \( n \)--descent if the algorithm is applied in a sufficient number of iterations to ensure \( n \) times a descent in the objective function. More formally, let \( n \) be a positive integer number and we denote

\[
f(\tilde{Z}) := \text{minimize} f(z)
\]

(4.5)

An \( n \)--descent iteration of \( A \) consists of applying the following algorithm:

\[
\begin{align*}
&\text{Let } \tilde{Z}' \text{ be the initial population and let } \ell = 0. \\
&\text{Do While } (\ell \leq n) \\
&\quad \tilde{Z} \in A(\tilde{Z}') \\
&\quad \text{If } f(\tilde{Z}') < f^* \text{ then } f^* = f(\tilde{Z}') \text{ and } \ell = \ell + 1 \\
&\quad \tilde{Z}' = \tilde{Z} \\
&\text{End Do While}
\end{align*}
\]
4.3. Instances of hybrid CG algorithms

The realization of a \( n \)-descent iteration is denoted by:

\[
\tilde{Z} \in \mathcal{A}(\tilde{Z}', n)
\]

(4.6)

\[\text{Figure 4.1: Hybridization of Algorithms}\]

In Table 4.1, we summarize the different steps of a CG algorithm belonging to the proposed framework. The resulting algorithm can be viewed as a hybrid algorithm of \( \mathcal{A}_c \) and \( \mathcal{A}_r \).

4.3 Instances of hybrid CG algorithms

Classical algorithms used in differentiable optimization such as SD, RSD or NSD Garcia et al. (2003) are roughly obtained letting \( Z \) be a point instead of a population, \( X \) a polyhedral set and \( X' \) as the convex hull of retained columns. These methods can be interpreted as a hybridization of \( \mathcal{A}_c = \) Frank-Wolfe and \( \mathcal{A}_r = \) (projected) Newton method.

In free derivatives optimization the line-search based modification of the Hooke and Jeeves method Hooke and Jeeves (1961) combine, in one iteration, a coordinate-wide search through each of the variables \( \mathcal{A}_c \) with a pattern search \( \mathcal{A}_r \). The \( \mathcal{A}_r \) produces an acceleration in the convergence of \( \mathcal{A}_c \).

In this Chapter we investigate numerically only the basic framework which consists of letting \( X = \mathbb{R}^n \), \( \hat{X}^t = \mathbb{R}^N \), \( Y^t = \{y^t\} \), \( \hat{X}^{t+1} = \{y^t\} \) and \( \tilde{Y}^{t+1} = \tilde{Y}^{t-1} \). Roughly, this basic hybrid algorithm consists of interchanging both algorithms when an \( n \)-descent iteration is carried out. We have introduced the following improvement. In order to take into account the random walking of \( \mathcal{A}_c \), the objective function \( f(x) \) in RMP is represented by \( f(d^t, x^T) \) where \( T \) is the transpose of a vector, and \( d^t = y^t - x^{t-1} \). This new function is a scalarization of the function \( f(x) \).

Some examples applied to this framework are explained below. The
1. (Initialization). Let \( \{ n_t^c \} \) and \( \{ n_t^r \} \) be two sequences of positive integer numbers. Let \( \tilde{Y}^1 \) and \( \tilde{X}^1 \) be respectively the initial populations for the algorithms \( A_c \) and \( A_r \). Let \( X^1 := \{ \emptyset \} \) and \( t := 1 \).

2. (Column Generation Algorithm). Apply a \( n_t^c \)-descent with the algorithm \( A_c \) on the problem \( P(f,X) \), starting from \( \tilde{Y}^t \). Let \( \tilde{Y}^t \) be the resulting population i.e., \( \tilde{Y}^t \in A(\tilde{Y}^t, n_t^c) \) and let \( y^t \) be the current best solution. i.e.,
\[
y^t := \arg\min_{y \in \tilde{Y}^t} f(y) \tag{4.7}
\]

3. (Set augmentation). Choose a set of columns \( \tilde{Y}^t \) satisfying \( y^t \in \tilde{Y}^t \subset \tilde{Y}^t \). Let \( X^{t+1} \subset X \) a nonempty closed set containing \( \{ \tilde{Y}^t, \tilde{X}^t \} \).

4. (Update of populations for RMP\( (f, \tilde{X}) \)). Let \( \tilde{X}^{t+1} \subset \tilde{Y}^t \cup \tilde{X}^t \).

5. (Restricted Master Algorithm). Apply a \( n_t^r \)-descent with the algorithm \( A_r \) on \( \text{RMP}(f,X^{t+1}) \), starting from \( \tilde{X}^{t+1} \). Let \( \tilde{X}^{t+1} \in A(\tilde{X}^{t+1}, n_t^r) \) be the resulting population. Let \( x^{t+1} \) be the current best solution. i.e.,
\[
x^{t+1} := \arg\min_{x \in \tilde{X}^{t+1}} f(x) \tag{4.8}
\]

6. (Update of population for \( P(f,X) \)). Choose a set of solutions \( \tilde{X}^{t+1} \) satisfying \( x^{t+1} \in \tilde{X}^{t+1} \subset \tilde{X}^{t+1} \) and update the population for solving \( P(f,X) \) as \( \tilde{Y}^{t+1} \subset \tilde{Y}^t \cup \tilde{X}^{t+1} \).

6. (Termination criterion ). If \( x^{t+1} \in \text{SOL}(f,X) \) → Stop. Otherwise, let \( t := t + 1 \). Go to Step 2.

### Table 4.1: Hybrid CG algorithm

Descriptions of this algorithms have been taken from Lopez-García et al. (2014a).

#### 4.3.1 Instances for \( A_r \)

**Nelder-Mead (NM)**

In the numerical experiments, for \( A_r \) the Nelder-Mead simplex search method Powell (1973) McKinnon (1973) has been used. This method is a direct search method that does not use numerical or analytic gradients and has local convergence with a high exploitation capacity. This algorithm described in Lagarias et al. (1999) uses a simplex of \( N + 1 \) points for \( N \)-dimensional vectors \( x \). The algorithm first makes a simplex around the initial point \( y^t \). Then, the algorithm modifies the simplex repeatedly generating at each step of the search a new point in or near the current simplex. The objective function value at the new point is compared with the values at the vertices of the simplex and, usually, one of the vertices is replaced by the new point, creating a new simplex.
4.3. Instances of hybrid CG algorithms

The Nelder-Mead pseudo-code is described in Table 4.2.

1. (Order) according to the values at the vertices: \( f(x_1) \leq f(x_2) \leq \cdots \leq f(x_{n+1}) \)
2. Calculate \( x_o \), the center of gravity of all points except \( x_{n+1} \).
3. (Reflection)
   - Compute reflected point \( x_r = x_o + \alpha (x_o - x_{n+1}) \)
     3.1. If the reflected point is better than the second worst, but not better than the best, i.e. \( f(x_1) \leq f(x_r) < f(x_n) \), then obtain a new simplex by replacing the worst point \( x_{n+1} \) with the reflected point \( x_r \), and go to step 1.
4. (Expansion)
   4.1. If the reflected point is the best point so far, \( f(x_r) < f(x_1) \), then compute the expanded point \( x_e = x_o + \gamma (x_o - x_{n+1}) \)
      4.1.1. If the expanded point is better than the reflected point, \( f(x_e) < f(x_r) \) then obtain a new simplex by replacing the worst point \( x_{n+1} \) with the expanded point \( x_e \), and go to step 1.
      4.1.2. Else obtain a new simplex by replacing the worst point \( x_{n+1} \) with the reflected point \( x_r \), and go to step 1.
4.2. Else (i.e. reflected point is not better than second worst) continue at step 5.
5. (Contraction)
   - Here, it is certain that \( f(x_r) \geq f(x_n) \)
     Compute contracted point \( x_c = x_o + \rho (x_o - x_{n+1}) \)
     5.1. If the contracted point is better than the worst point, i.e. \( f(x_c) < f(x_{n+1}) \) then obtain a new simplex by replacing the worst point \( x_{n+1} \) with the contracted point \( x_c \), and go to step 1.
5.2. Else go to step 6.
6. (Reduction)
   - For all \( i \in \{2, \ldots, n+1\} \) replace the point with \( x_i = x_1 + \sigma (x_i - x_1) \) and go to step 1.

Note: \( \alpha, \gamma, \rho \) and \( \sigma \) are respectively the reflection, the expansion, the contraction and the shrink coefficient. Standard values are \( \alpha = 1, \gamma = 2, \rho = -1/2 \) and \( \sigma = 1/2 \).

| Table 4.2: Nelder-Mead algorithm pseudocode |
4.3.2 Instances for $A_c$

On the other hand, for $A_c$, two algorithms with properties of global convergence have to be employed. The motivation is to combine global convergent methods with a local convergent method which is more computationally efficient. In particular in this Chapter Simulated Annealing and Particle Swarm Optimization algorithms have been used for $A_c$.

**Simulated Annealing (SA)**

Simulated annealing is a popular local search metaheuristic. The key feature of simulated annealing is that it provides a means to escape local optima by allowing hill-climbing moves in hopes of finding a global optimum.

Simulated annealing starts with an initial solution $x$. At each iteration of a simulated annealing algorithm the current solution $x$ and a newly selected solution $x'$ are compared. The new solution is generated (either randomly or using some pre-specified rule) in a neighborhood $N(x)$ of the current solution $x$. The candidate solution $x'$ is accepted based on the rule

$$P(\text{Accept } x' \text{ as next solution}) = \begin{cases} \exp\left[-\frac{(f(x') - f(x))}{T_n}\right] & \text{if } f(x') - f(x) > 0 \\ 1 & \text{if } f(x') - f(x) \leq 0 \end{cases}$$

Improving solutions are always accepted, while a fraction of non-improving solutions are accepted in the hope of escaping local optima in search of global optima. The probability of accepting non-improving solutions depends on a temperature parameter $T_n$.

Simulated annealing is outlined in Table 4.3. The SA algorithm used in this Chapter was proposed by Bohachevsky et al. (1986) is an improved version of the classical SA Van Laarhoven and Aarts (1987).

**Particle Swarm Optimization (PSO)**

The second algorithm used is the original PSO algorithm using the global gbest model. It is a population-based algorithm and evolutionary in nature, introduced by Kennedy and Eberhart (1995). PSO is a kind of random search algorithm based on the metaphors of social interaction and communications. This method has been shown to be effective in solving difficult and complex optimization problems in a wide range of fields Poli (2008) Angulo et al. (2011). PSO maintains at each iteration a set of swarm particles, feasible points represented by $(\tilde{Y}')$ in our framework.

The PSO starts with a random population of solutions (particles) in the search space. In every iteration, each particle is updated by following two “best” values. The first, the so called pbest, is the best solution that is found by the particle. The second, the so called gbest, is the best solution
4.3. Instances of hybrid CG algorithms

1. (Initialization). Initialize the number of iterations \(N\). Select an initial solution \(x\), a temperature cooling schedule \(\{T_n\}\), and an initial temperature \(T_0\). Select a repetition schedule, \(\{M_n\}\), that defines the number of iterations executed at each temperature. Set the temperature change counter \(n = 0\) and repetition counter \(m = 0\).

1. Generate a solution \(x' \in N(x)\).
2. Calculate \(\Delta = f(x') - f(x)\). If \(\Delta \leq 0\) then \(x = x'\), \(f^* = f(x')\) and \(x^* = x'\). Otherwise \(\Delta > 0\), set \(x = x'\) with probability \(\exp[-\Delta/T_n]\). Take \(m = m + 1\).
3. If \(m = M_n\) then \(n = n + 1\) and \(m = 0\).
4. (Stopping criterion). If the current number of iterations is \(t = N\), Stop; otherwise let \(t = t + 1\) and go to 1.

Output: The best solution found \(x^*\) and its objective value \(f^*\).

Table 4.3: SA algorithm

that is found by whole swarm members. After finding the two best values, the position of each particle at iteration \(t\) will be updated by the following equations:

\[
v^t_{pk} = v^{t-1}_{pk} + c_1 \cdot \text{Rand}() \cdot (p^{t-1}_{pk} - x^{t-1}_{pk}) + c_2 \cdot \text{Rand}() \cdot (g^{t-1}_{pk} - x^{t-1}_{pk}) \quad (4.9)
\]
\[
x^t_{pk} = x^{t-1}_{pk} + v^t_{pk} \quad (4.10)
\]

\(v^t_{pk}\) is the \(k\)-component of the velocity vector of the particle \(p\) at the iteration \(t\); the current values of \(p_{\text{best}}\) and \(g_{\text{best}}\) are \(p^{t-1}_{pk}\) and \(g^{t-1}_{pk}\); \(\text{Rand}()\) is a random number in \([0, 1]\); \(c_1\) is a learning factor; \(c_2\) is a social learning factor; and finally \(x^t_{pk}\) is the \(k\)-component of new position vector of the particle \(p\) at the iteration \(t\). The velocity \(v^t_{pk}\) belongs to \([-V_{\text{max}}, V_{\text{max}}]\), where \(V_{\text{max}}\) is a designated maximum velocity. If the velocity on one dimension exceeds the maximum, it will be set to \(V_{\text{max}}\).

The original version has been improved in three aspects. The first improvement is the introduction of inertia weight \(\omega\) (Shi and Eberhart (1998)) or equivalently the constraint factor (Clerc and Kennedy (2002)). These parameters was developed to better balance exploration and exploitation phases and they avoid the use of \(V_{\text{max}}\) which was viewed as both artificial and difficult to balance. The second one called LBEST version considers, in order to avoid a premature convergence of the algorithm, each particle keeps track of the best solution, called \(l_{\text{best}}\), attained within a local topological neighbourhood of particles instead to considerer the overall best value in the whole particle swarm. The third modification is concerned with velocity updated rule. The original version chooses at random (uniform distribution) point inside hyperparallelepiped. Standard PSO 2011 (Zambrano-Bigiarini
et al. (2013) propose the make use of sampling in hyperspheres to update the velocity. Table 4.4 shows the standard PSO algorithm.

1. **(Initialization).** Initialize the number of iterations (N), the number of particles (X), learning factor (c), the inertia weight (w) and the average number of informants (f). All parameters must be determined according to the variables of the problem. Set \( f_p^{vb} = f_p^{lb} = +\infty \) for all particle p. Initialize particles with random position and velocity. Let \( N_p^t \) be the set of neighbours of the particle p at iteration \( t = 1 \).

2. **(Evaluation).** Evaluate the objective function for each particle p.

   \[
   f_p^{t-1} = f(x_p^{t-1}) \quad \text{with} \quad p = 1, \cdots, N
   \]

   (4.11)

3. **(Select the 'pbest').** Find the best value for each particle p. If \( f_p^{t-1} \) is better than their previous values, then the current local minimum is \( pb_p^{t-1} = x_p^{t-1} \) as the new 'pbest' of the particle p and \( J_p^{t-1} = f_p^{t-1} \).

4. **(Select the 'lbest' in the neighborhood).** Update the best of the best positions found up to now by informants as follows: Let \( p' = \arg \min_{q \in N_p^{t-1}} \{ f_q^{p_b} \} \). If \( f_p^{p_b} \) is lesser than the current value \( f_p^{t-1} \), then \( f_p^{p_b} = f_p^{t-1} \) and \( lb_p^{t-1} = x_p^{t-1} \).

5. **(Update velocity and position of the particles).** If \( lb_p^{t-1} \neq pb_p^{t-1} \) then set \( c_t^{p} = c \); otherwise \( c_t^{p} = \frac{3}{N} \). Let \( G_p^{t-1} = x_p^{t-1} + c_t^{p} \cdot \left( lb_p^{t-1} + \frac{3}{8} \cdot J_p^{t-1} \right) \) and let \( \bar{x}_p^{t-1} \) be a random point in the hyper-sphere with center \( G_p^{t-1} \) and radius \( ||x_p^{t-1} - G_p^{t-1}|| \).

   5.1. Update the particle velocity for each particle p according to the following rule

   \[
   v_p^{t} = \omega v_p^{t-1} + \bar{x}_p^{t-1} - x_p^{t-1}
   \]

   (4.12)

   5.2. The updating rule for the particle position is as follows:

   \[
   x_p^{t} = x_p^{t-1} + v_p^{t}
   \]

   (4.13)

6. **(Stopping criterion).** If the current iteration number is \( t = N \), Stop; otherwise continue. If the current iteration is unsuccessful (no improvement of the best objective value), define new neighborhoods \( N_p^t \) (refer the reader to Zambrano-Bigiarini et al. (2013) for a description of the adaptive random topology); otherwise \( N_p^t = N_p^{t-1} \). Let \( t = t + 1 \) and go to 2.

   **Output:** The best solution found defined by \( lb_p^{N-1} \) where \( p' = \arg \min_{p} \{ f_p^{p_b} \} \) and its objective value \( f_p^{p_b} \).

**Table 4.4:** SPSO algorithm

### 4.3.3 Instances of the framework

The algorithms presented below shows some drawbacks related to their nature. SA and PSO are centred in exploration and NM is centred in exploitation.

Exploration is related to global search as well as exploitation is related to
4.4 Computational results

local search in these algorithms. The intention of the first one is to explore the search space looking for good solutions, whereas the second one tries to refine the solution and avoid big jumps on the search space.

A hybridization of the shown algorithms is proposed in order to combine their features. This area is interesting due to its versatility in optimization problems. The resulting method is a derivative free algorithm.

To find a better solution SA and PSO are used in an exploration phase until nc improvements are obtained. Then NM is used in an exploitation phase, coming out nr iterations and finally updating the best solution found with the solution of NM stage.

Figure 4.2 shows how the PSO+NM algorithm works. The SA+NM implementation is similar to PSO+NM but it does not consider a population, only a particle.

**Figure 4.2:** PSO+NM scheme.

### 4.4 Computational results

In this section the framework proposed in this Chapter is tested by the hybridization of SA with NM (SA+NM) and PSO with NM (PSO+NM) to test the improvement of the algorithms used. As mentioned above, the main
motivation of this hybridization is to combine the capacity of the SA and PSO algorithms to find a global minimum with the capacity of the NM algorithm to obtain a more precise solution near a local minimum.

In the next sections, the design of experiments is explained, and the results are reported comparing our approach with the SA, NM and PSO algorithms.

4.4.1 Design of the experiments

To compare the quality of the proposed modified algorithms it is necessary to test them with a large-scale test process on a variety of response functions, ignoring the possible effectiveness of the algorithm in a specific type of function.

The set of 20 deterministic functions used in the computational experiments are found in Fan and Zahara (2007) and described in Appendix A, and have been selected because they are a set of curvilinear functions for difficult unconstrained optimization problems with a variety of dimensions \(N\) and functional forms, which makes it possible to assess the robustness of the proposed approach.

To evaluate the algorithms’ effectiveness each of them have been executed 100 times per test function. In each of the executions, the initial point in SA, NM and SA+NM is sampled from Uniform(-50,50) and initial particles in PSO and PSO+NM are randomly generated from Uniform(-50,50). Also, in PSO and PSO+NM algorithms the number of particles is defined as \(5N\).

The stopping criterion selected for all the algorithms is that the current execution reaches 5000\(N^2\) function evaluations, which corresponds to 1000\(N\) iterations of the PSO algorithm. The sequence of number of descents was \(\{n_t^c\} = 5\) and \(\{n_t^r\} = 100\) in the numerical tests.

The criteria selected to compare the algorithm’s robustness, effectiveness, efficiency and accuracy using the results obtained in the experiments described above are:

1. Rate of successful minimizations, considering that a successful minimization occurs when \(|f(X^e) - f^*| < 10^{-10}\) where \(f^* = \min\{f(x) : x \in X\}\).

2. Average of function evaluations until a successful minimization occurs.

3. Gap between the best minimum found in the 100 executions and the optimum of the function.
4.4. Computational results

4.4.2 Results

The results given in this section will show that the approach described is capable of improving the SA, NM and PSO algorithms.

Tables 4.5 and 4.6 show the complete results of the experiments performed. By comparing them, it can be seen that the modified version of the algorithms in general has a significantly higher rate of successful minimization and a lower average of objective function evaluation before reaching a successful minimization than the original algorithms.

To prove these statements the non-parametric statistical Wilcoxon test is carried out with a significance of 0.05 to compare the paired groups SA vs SA+NM, NM vs SA+NM, PSO vs PSO+NM and NM vs PSO+NM, and gives the results shown in Table 4.7. Taking into account these results, it can be stated that there exists a clear improvement in the number of successful minimizations for all the algorithms.

In the case of function evaluations until a successful minimization occurs, the results show that in NM and SA vs SA+NM there are no conclusive results because of the lack of successful minimizations in the NM and SA algorithms, but in PSO and NM vs PSO+NM it can be seen that there does exist an improvement, reducing the number of evaluations necessary to find the minimum of the function.

To illustrate the general performance of the approaches in terms of evaluations of the objective function, Figure 4.3 show the 2nd function of the experiments, (B2 function), plotting the best evaluation of the function vs the number of evaluations of the objective function. It can be seen from the figures that the SA+NM and PSO+NM algorithms converge more quickly than the classical algorithms, also breaking the local minimum which cannot be reduced by the NM algorithm.

![Figure 4.3: Objective Function vs Evaluations for SA, NM and SA+NM](image)

Finally, for the gap with the best found minimum results it can be shown
chapter 4. a framework for derivative free algorithm

hybridization

that the SA is improved by the SA+NM algorithm and the NM is improved by the PSO+NM algorithm.

4.5 Conclusions

In this Chapter a new framework for hybridization of derivative free algorithms is presented. This approach includes the hybridization of two methods in an attempt to combine the specific capacity of exploitation and exploration of each algorithm, increasing the robustness, effectiveness, efficiency and accuracy.

This Chapter investigates numerically the basic CG algorithm consisting of letting $\hat{X} = X = \mathbb{R}^N$ and the algorithms NM, SA and PSO. Also a set of computational tests has been done using 20 curvilinear functions for difficult unconstrained optimization, in which the original algorithms are compared with the modified versions. The results show that the hybrid algorithms have a higher rate of successful minimizations and an acceleration of the algorithms, which are capable of giving the optimum using fewer evaluations of the objective function than the original algorithm.

Future work will research new instances of the algorithm based on the initialization Fan and Zahara (2007) or augmentation rules for defining $\hat{X}$. The idea is to extend the computational results by trying to apply the framework to other algorithms.

This approach will be used in Chapters 5 and 6 for solving more complex optimization problems.
### 4.5. Conclusions

Successful minimizations

<table>
<thead>
<tr>
<th>Function</th>
<th>NM</th>
<th>SA</th>
<th>SA+NM</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>17</td>
<td>0</td>
<td>17</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>24</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>0</td>
<td>46</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>97</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0</td>
<td>87</td>
</tr>
<tr>
<td>7</td>
<td>69</td>
<td>5</td>
<td>64</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>0</td>
<td>29</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>0</td>
<td>23</td>
</tr>
<tr>
<td>12</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>13</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>14</td>
<td>0</td>
<td>0</td>
<td>29</td>
</tr>
<tr>
<td>15</td>
<td>78</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>16</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>17</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>18</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>19</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>20</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Function Evaluations

<table>
<thead>
<tr>
<th>Function</th>
<th>NM</th>
<th>SA</th>
<th>SA+NM</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.829e+02</td>
<td>-</td>
<td>1.479e+04</td>
</tr>
<tr>
<td>2</td>
<td>-</td>
<td>-</td>
<td>1.816e+04</td>
</tr>
<tr>
<td>3</td>
<td>2.912e+02</td>
<td>-</td>
<td>1.450e+04</td>
</tr>
<tr>
<td>4</td>
<td>-</td>
<td>-</td>
<td>1.658e+04</td>
</tr>
<tr>
<td>5</td>
<td>-</td>
<td>-</td>
<td>4.504e+04</td>
</tr>
<tr>
<td>6</td>
<td>-</td>
<td>-</td>
<td>4.480e+04</td>
</tr>
<tr>
<td>7</td>
<td>4.014e+02</td>
<td>2.757e+04</td>
<td>2.210e+04</td>
</tr>
<tr>
<td>8</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>9</td>
<td>-</td>
<td>-</td>
<td>7.993e+04</td>
</tr>
<tr>
<td>10</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>11</td>
<td>-</td>
<td>-</td>
<td>8.001e+04</td>
</tr>
<tr>
<td>12</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>13</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>14</td>
<td>-</td>
<td>-</td>
<td>3.200e+05</td>
</tr>
<tr>
<td>15</td>
<td>3.499e+03</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>16</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>17</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>18</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>19</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>20</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Gap with best minimum found

<table>
<thead>
<tr>
<th>Function</th>
<th>NM</th>
<th>SA</th>
<th>SA+NM</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.769e-13</td>
<td>2.190e-08</td>
<td>2.827e-15</td>
</tr>
<tr>
<td>2</td>
<td>1.411e-09</td>
<td>1.035e-05</td>
<td>3.919e-12</td>
</tr>
<tr>
<td>3</td>
<td>4.300e-11</td>
<td>1.191e-06</td>
<td>1.943e-13</td>
</tr>
<tr>
<td>4</td>
<td>3.756e-10</td>
<td>7.366e-07</td>
<td>9.437e-16</td>
</tr>
<tr>
<td>5</td>
<td>2.181e-10</td>
<td>1.885e-04</td>
<td>4.288e-11</td>
</tr>
<tr>
<td>6</td>
<td>3.941e-10</td>
<td>6.979e-06</td>
<td>2.553e-13</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>2.208e-12</td>
<td>1.449e-16</td>
</tr>
<tr>
<td>8</td>
<td>3.169e-10</td>
<td>3.652e-04</td>
<td>1.167e-07</td>
</tr>
<tr>
<td>9</td>
<td>4.480e-10</td>
<td>8.617e-05</td>
<td>2.130e-12</td>
</tr>
<tr>
<td>10</td>
<td>1.261e-10</td>
<td>1.809e-03</td>
<td>5.536e-09</td>
</tr>
<tr>
<td>11</td>
<td>9.319e-10</td>
<td>6.986e-05</td>
<td>3.390e-13</td>
</tr>
<tr>
<td>12</td>
<td>6.552e-08</td>
<td>3.071e-06</td>
<td>1.173e-06</td>
</tr>
<tr>
<td>13</td>
<td>5.403e-07</td>
<td>3.247e-06</td>
<td>1.175e-06</td>
</tr>
<tr>
<td>14</td>
<td>7.027e-10</td>
<td>1.916e-03</td>
<td>6.095e-12</td>
</tr>
<tr>
<td>15</td>
<td>1.704e-14</td>
<td>6.399e-03</td>
<td>4.059e-06</td>
</tr>
<tr>
<td>16</td>
<td>1.009e-01</td>
<td>6.264e-01</td>
<td>5.793e-01</td>
</tr>
<tr>
<td>17</td>
<td>2.358e+02</td>
<td>5.081e+00</td>
<td>1.990e+00</td>
</tr>
<tr>
<td>18</td>
<td>2.949e+00</td>
<td>1.406e+00</td>
<td>1.686e+00</td>
</tr>
<tr>
<td>19</td>
<td>2.544e+00</td>
<td>1.406e+00</td>
<td>1.686e+00</td>
</tr>
<tr>
<td>20</td>
<td>7.227e-01</td>
<td>4.031e+00</td>
<td>4.348e+00</td>
</tr>
</tbody>
</table>

Table 4.5: NM, SA and SA+NM Results
| Function | NM | PSO | PSO+NM | NM | PSO | PSO+NM | NM | PSO | PSO+NM | NM | PSO | PSO+NM | NM | PSO | PSO+NM | NM | PSO | PSO+NM |
|----------|----|-----|--------|----|-----|--------|----|-----|--------|----|-----|--------|----|-----|--------|----|-----|--------|----|-----|--------|----|-----|--------|----|-----|--------|----|-----|--------|----|-----|--------|
| 0 0 10 | 0 0 17 | 48 64 | 1.829e+02 | 0 0 13 10 | 0 0 0 0 | 10 20 | 0 36 | 22 |
| 0 0 0 0 | 0 0 17 | 48 64 | 1.829e+02 | 0 0 13 10 | 0 0 0 0 | 10 20 | 0 36 | 22 |
| 1 7 | 1 4 | 1.883+08 | 12 914.7 | 1.478e+02 | 1 8 | 3 | 3 | 0 0 | 18 | 11 19 | 44 44 | 0 0 |
| 0 0 0 0 | 0 0 10 | 0 0 17 | 48 64 | 1.829e+02 | 0 0 13 10 | 0 0 0 0 | 10 20 | 0 36 | 22 | 0 0 0 0 |
| 1 7 | 1 4 | 1.883+08 | 12 914.7 | 1.478e+02 | 1 8 | 3 | 3 | 0 0 | 18 | 11 19 | 44 44 | 0 0 |
| 0 0 0 0 | 0 0 10 | 0 0 17 | 48 64 | 1.829e+02 | 0 0 13 10 | 0 0 0 0 | 10 20 | 0 36 | 22 | 0 0 0 0 |
| 1 7 | 1 4 | 1.883+08 | 12 914.7 | 1.478e+02 | 1 8 | 3 | 3 | 0 0 | 18 | 11 19 | 44 44 | 0 0 |
| 0 0 0 0 | 0 0 10 | 0 0 17 | 48 64 | 1.829e+02 | 0 0 13 10 | 0 0 0 0 | 10 20 | 0 36 | 22 | 0 0 0 0 |
| 1 7 | 1 4 | 1.883+08 | 12 914.7 | 1.478e+02 | 1 8 | 3 | 3 | 0 0 | 18 | 11 19 | 44 44 | 0 0 |
| 0 0 0 0 | 0 0 10 | 0 0 17 | 48 64 | 1.829e+02 | 0 0 13 10 | 0 0 0 0 | 10 20 | 0 36 | 22 | 0 0 0 0 |
| 1 7 | 1 4 | 1.883+08 | 12 914.7 | 1.478e+02 | 1 8 | 3 | 3 | 0 0 | 18 | 11 19 | 44 44 | 0 0 |
| 0 0 0 0 | 0 0 10 | 0 0 17 | 48 64 | 1.829e+02 | 0 0 13 10 | 0 0 0 0 | 10 20 | 0 36 | 22 | 0 0 0 0 |
| 1 7 | 1 4 | 1.883+08 | 12 914.7 | 1.478e+02 | 1 8 | 3 | 3 | 0 0 | 18 | 11 19 | 44 44 | 0 0 | 0 0 0 0 | 20 30 11 10 | 22 22 22 22 | 0 0 0 0 | 10 20 | 0 36 | 22 |

Table 4.6: NM, PSO and PSO+NM Results
4.5. Conclusions

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>S. minimizations</th>
<th>F. Evaluations</th>
<th>Gap best minimum</th>
</tr>
</thead>
<tbody>
<tr>
<td>NM vs SA+NM</td>
<td>0.037</td>
<td>0.0545*</td>
<td>0.3005</td>
</tr>
<tr>
<td>SA vs SA+NM</td>
<td>0.0025</td>
<td>-*</td>
<td>0.0285</td>
</tr>
<tr>
<td>NM vs PSO+NM</td>
<td>0.00015</td>
<td>0.034</td>
<td>0.0001</td>
</tr>
<tr>
<td>PSO vs PSO+NM</td>
<td>0.001</td>
<td>0.0001</td>
<td>0.2345</td>
</tr>
</tbody>
</table>

* Not enough data

**Table 4.7:** Significances of Wilcoxon tests
5. Constrained logit model: Formulation and calibration
A constrained logit-type choice (CMNL) model is presented in which, using the entropy-maximizing framework, a cut-off approach is considered. Moreover, Reproducing Kernel Hilbert Spaces are used to deal with (dynamic) non-linear utilities. The objective is to represent simultaneously the dynamic aspect and the restrictions in the selection process. A novel calibration procedure is introduced in which the utilities are viewed as the parameters of the proposed model instead of attribute weights as in the classical linear models. A discussion on over-specification of the CMNL model is presented.

This methodology is illustrated with a railway service choice problem in which users choose the service they will use depending on the timetable, ticket price, travel time and seat availability (capacity constraints). A hybridization of the Standard Particle Swarm Optimization and Nelder-Mead methods has been applied for the calibration, and the viability of the proposed approach is shown in a real case study in Spain.

5.1 Introduction

Discrete choice models have long been recognized for their ability to capture a broad range of transport-related choice phenomena. However, certain applications require that we simultaneously tackle a specific set of constraints and time-of-day (TOD) modelling explicitly within the choice process. Taking this into account, this Chapter proposes a random utility framework to model both issues.

Discrete choice random utility models assume a compensatory mechanism in which the consumer’s strategy imposes a trade-off between at-
CHAPTER 5. CONSTRAINED LOGIT MODEL: FORMULATION AND CALIBRATION

attributes. A more realistic approach requires to relax the compensatory assumption and to cope with constraints related to individual’s behaviour.

Theoretical frameworks have been proposed to relax this assumption. A rough taxonomy classifies the approaches followed in the literature into hard or soft methods. These methods are called hard in the sense that the cut-offs which are imposed on choice makers cannot be violated for a valid choice. Manski (1977), Swait and Ben-Akiva (1987), Ben-Akiva and Boccara (1995), Cantillo and Ortúzar (2005), Cantillo et al. (2006), among others, use a two-stage approach to model the choice set generation process. In the first stage, the alternatives are screened by some non-compensatory process and in the second stage, a compensatory model calculates the choice probability conditioned by the remaining alternatives.

The so-called soft methods incorporate thresholds for perception in attribute values implicitly.

Swait (2001) incorporates attribute cut-offs into the utility maximization problem formulation explicitly. It makes it possible for the consumer to treat the constraints as soft by violating them at some cost.

Cascetta and Papola (2001) propose the implicit availability/perception model (IAP). The choice-set of alternatives is a fuzzy set where each element has a degree of membership of the choice set.

Martínez et al. (2009) formulates a constrained multinomial logit (CMNL). This model implements cut-offs as a binomial logit function embedded in multinomial logit models. The MNL model allows the choice domain to be constrained by as many cut-offs as required, limiting both an upper and a lower level of variables.

Anas (1983) demonstrated that information minimizing (or entropy-maximizing modelling) and utility maximizing (behavioural demand modelling) should be seen as two equivalent views of the same problem. Donoso and de Grange (2010) give an interpretation of the entropy maximization problem in the context of microeconomic modelling, attempting to explain the origin of the two problems’ equivalence.

In this Chapter we adopt the entropy-maximizing approach adding the non-linear constraints into this formulation forcing the constraints to become hard. The main difference between our entropy maximization problem and those presented in Anas (1983), Donoso and de Grange (2010), De Grange et al. (2013) is that general utilities are considered, not necessarily having linear attributes. This issue is essential to dynamic choice modelling, such as the departure time modelling which is the second goal of the Chapter.

Srinivasan and Mahmassani (2005) adopt the kernel logit model, also referred to as mixed-logit model by several authors, to model longitudinal discrete choice data. This model considers that a decision maker faces the decision of selecting an alternative from a set of alternatives and this
decision process is being respected over $T$ time periods. This framework accommodates a flexible covariance structure to both alternatives over time, and it is capable of representing unobservable results from general dynamic and stochastic processes. In contrast to these authors, this work considers only the selection of a unique alternative in a given time interval.

Popuri et al. (2008) introduced continuous (dynamic) systematic utility functions in their models, via sinusoidal functions which interact with covariates. In this work the most general framework, Reproducing Kernel Hilbert Spaces (RKHS), is considered, introducing dynamic non-linear utilities. The main advantage of this approach is that the modelling is not centred on a specific functional form (linear, sinusoidal, etc) but on belonging to a specific Hilbert space defined by the so-called Mercer kernel. In this Chapter a new method for calibration is developed based on the novel point of view of considering a subset of utilities as parameters for the estimation instead of the classical weightings of the attributes. This approach obtains the multi-attribute utility functions and they are used to calculate all of the un-estimated utilities. This is possible due to the current computational advantages which allow working with free derivative optimization methods. This fact allows us to use functions implicitly defined by means of an optimization model.

The above modelling aspects, i.e. the inclusion of constraints in the decision space and the use of utilities depending on time, have been illustrated with the modelling of the selection of railway services in which the capacity of the train fleet imposes various constraints on the set of available services. Furthermore users have preferences as to which train they select depending on the departure time, price and travel time. This model has been dealt with by a constrained nested logit model. The calibration of the model has been illustrated numerically.

The Chapter is organised as follows. Section 5.2 formulates the constrained logit model. In Section 5.3 Reproducing Kernel Hilbert Spaces are defined to represent generic utility functions. Furthermore the Tikhonov Regularization Theory method is explained for the estimation of these functions. Section 5.4 discusses the procedure followed to estimate the constrained nested logit with this type of non-linear utility and the over-specification problems. Section 5.2.4 illustrates the calibration by numerically solving a railway service selection problem. Finally Section 5.6 concludes with a discussion of our findings and future work.

5.2 Formulation of the constrained logit model

In this section the constrained logit model is formulated. The first step demonstrates that the maximum entropy approach and maximum expected utility are equivalent for the logit model case (Gumbel errors). Next, this scheme is utilized to introduce constraints in the users' individual decision-
making processes and subsequently homogeneous groups of users are considered. Finally this approach is extended to the constrained nested logit models.

5.2.1 Formulation of the multinomial logit model as a maximum entropy optimization problem

The entropy maximization problem leads to the multinomial logit model (see Anas (1983), Donoso and de Grange (2010), De Grange et al. (2013)). In this section this approach has been used, considering the indirect utility function of individual \( i \) conditioned upon choosing alternative \( m \) without limiting the model to a linear relationship between attributes.

\[
\begin{align*}
\text{minimize} & \quad \sum_i \sum_m \left[ \frac{1}{\lambda_i} p_i^m (\ln p_i^m - 1) - p_i^m V_i^m \right], \\
\text{subject to:} & \quad \sum_m p_i^m = 1; \forall i(\Phi_i) 
\end{align*}
\]

(5.1)-(5.2)

where \( p_i^m \) is the probability that individual \( i \) choose alternative \( m \), \( V_i^m \) represents the deterministic part of the indirect utility perceived by individual \( i \) conditional upon choosing alternative \( m \), and \( \lambda_i \) is a scalar associated with the variance of the error term of the utilities.

Applying the optimality conditions to problem (5.1)-(5.2)

\[ \frac{1}{\lambda_i} \ln p_i^m - V_i^m - \Phi_i = 0 \]

(5.3)

Solving for \( p_i^{m'} \) in (5.3) and summing over \( m \), we get

\[
\begin{align*}
p_i^{m'} &= \exp \left\{ \lambda_i \left( V_i^{m'} + \Phi_i \right) \right\} = \exp(\lambda_i \Phi_i) \exp(\lambda_i V_i^{m'}) \\
1 &= \sum_m p_i^m = \exp(\lambda_i \Phi_i) \sum_m \exp(\lambda_i V_i^m) 
\end{align*}
\]

(5.4)-(5.5)

Finally, dividing (5.4) by (5.5) we get

\[
p_i^{m'} = \frac{\exp(\lambda_i V_i^{m'})}{\sum_m \exp(\lambda_i V_i^m)}
\]

(5.6)

The maximum expected utility (EMU) of an individual \( i \) in an MNL-type selection process (Williams (1977)) is

\[
EMU_i = \frac{1}{\lambda_i} \ln \sum_m \exp(\lambda_i V_i^m)
\]

(5.7)
5.2. Formulation of the constrained logit model

De Grange et al. (2013) infer

\[ EMU_i = \sum_m p_i^m V_i^m - \frac{1}{\lambda_i} \sum_m p_i^m \ln p_i^m. \]  

(5.8)

Moreover, considering that \( \sum_m p_i^m = 1 \)

\[-EMU_i - \frac{1}{\lambda_i} = -\sum_m p_i^m V_i^m + \frac{1}{\lambda_i} \sum_m p_i^m (\ln p_i^m - 1). \]  

(5.9)

It shows that maximizing the expected utility and minimizing the entropy are equivalent approaches.

5.2.2 The disaggregated constrained multinomial logit model

The above problem can be modified to tackle the consumer’s feasible domain by imposing the set of constraints explicitly on decisions

\[
\text{minimize} \quad \sum_i \sum_m \left[ \frac{1}{\lambda_i} p_i^m (\ln p_i^m - 1) - p_i^m V_i^m \right],
\]

subject to:

\[
\sum_m p_i^m = 1; \forall i (\Phi_i) \quad (5.11)
\]

\[
h_r(p) \leq b_r; \forall r (\eta_r) \quad (5.12)
\]

where \( p = (\ldots, p_i^m, \ldots) \) and \( h_r(\cdot) \) are convex functions for all \( r \).

An example of the above model is the polarized logit model (De Grange et al. (2013)) in which only one constraint is stated \( h(p) = \sum_{i,m} p_i^m (1-p_i^m) \leq \varepsilon \).

CMNL is a convex program with a strictly convex objective function which has a single solution. The Karush-Kuhn-Tucker optimality conditions lead to

\[
p_i^{m'} = \frac{\exp\{\lambda_i (V_i^{m'} + \sum_r \eta_r \partial h_r/\partial p_i^{m'})\}}{\sum_m \exp\{\lambda_i (V_i^{m} + \sum_r \eta_r \partial h_r/\partial p_i^{m})\}} \quad (5.13)
\]

Note that the dual variables \( \eta_r \leq 0 \) by KKT conditions. Moreover, if the constraints are inactive \( \eta_r = 0 \) for all \( r \) then the classical MNL model appears.

Eq. (5.13) shows that the constraints (5.12) impose a penalization to the utilities. Swait (2001), Martínez et al. (2009) penalize utilities if a choice is made outside the domain, defined by linear constraints obtaining the same result as the proposed model, a penalization of the utilities. The main
difference is that the penalizations of the proposed approach come from the imposed constraints, as opposed to in other models where they are fixed by users. Swait (2001) assumes a linear penalty and Martinez et al. (2009) the following non-linear cut-off term

\[ \eta_m^i = -\frac{1}{\lambda_i} \sum_r \ln \left[ 1 + \exp \left\{ w_r(h_r(p) - b_r + \rho_r) \right\} \right] \] (5.14)

where \( \rho_r = \frac{1}{\omega_r} \ln \left( \frac{1 - \psi_r}{\psi_r} \right) \) and \( \psi_r \) is the cut-off tolerance parameter which defines the probability at the boundary. These models perform a soft compliance with the constraints. The proposed model is applied for deterministic compliance of constraints. It is worth noting that in \( r \) the constraint is active

\[ \eta_r^m = -\frac{1}{\lambda_i} \ln(\psi_r) = \eta_r \partial h_r/\partial p_r^m = \eta_r \] (5.15)

This equation shows that an adequate value of \( \psi_r \) generates multiplier \( \eta_r \).

The advantage of the proposed approach is that \( \eta_r \) is determined by the constraint \( r \), whereas Martinez et al. (2009) selects \( \psi_r \) and \( \omega_r \) a priori as the main rule to be satisfied.

### 5.2.3 The aggregate constrained multinomial logit model (CMNL)

In some cases, the demand could only be represented in an aggregated way. Consequently data of individual user’s attributes cannot be collected. In this section the previous model is adapted to this case. The population is classified into a set of socioeconomic groups \( \mathcal{L} \), whose members are assumed to behave identically, except for random errors in individual decision-making \( \varepsilon_i \). Denote an element of \( \mathcal{L} \) as \( \ell \) and for \( \ell \) the set of individuals of type \( \ell \). Therefore \( V_m = V^{m\ell} \) and \( \lambda_i = \lambda^\ell \) for \( i \in \ell \). This assumption implies that \( p_i^m = p^{m\ell} \) for each \( i \in \ell \). Now the problem (5.10)-(5.12) will be reformulated using the aggregated variables \( g^{m\ell} \) which represent the number of individuals of type \( \ell \) who select alternative \( m \). The relationship between aggregated and disaggregated variables is described by:

\[ g^{m\ell} = \sum_{i \in \ell} p_i^m \] (5.16)
5.2. Formulation of the constrained logit model

Let $\hat{g}^\ell$ be the number of users of type $\ell$, thus the objective function is stated as:

$$
\sum_i \sum_m \left[ \frac{1}{\lambda_i} p_i^m (\ln p_i^m - 1) - p_i^m V_i^m \right] = \\
\sum_{i \in \mathcal{I}} \sum_{m} \sum_{\ell} \left[ \frac{1}{\lambda_i} p_i^m (\ln p_i^m - 1) - p_i^m V_i^m \right] = \\
\sum_{m, \ell} \left[ \frac{1}{\lambda} g^{m\ell} (\ln p^{m\ell} - 1) - g^{m\ell} V^{m\ell} \right] = \\
\sum_{m, \ell} \left[ \frac{1}{\lambda} g^{m\ell} \left( \ln \left( \frac{\hat{g}^{m\ell}}{\hat{g}^\ell} \right) - 1 \right) - g^{m\ell} V^{m\ell} \right] = \\
- \sum_{\ell} \hat{g}^\ell \ln \hat{g}^\ell + \sum_{m, \ell} \left[ \frac{1}{\lambda} g^{m\ell} (\ln g^{m\ell} - 1) - g^{m\ell} V^{m\ell} \right]
$$

Adding constraints (5.11) with respect to $i \in \mathcal{I}_\ell$, we obtain that:

$$
\sum_{i \in \mathcal{I}_\ell} \sum_{m} p_i^m = \sum_{m} g^{m\ell} = \hat{g}^\ell 
$$

Finally, we will rewrite the constraints (5.12) with respect to the aggregated variable $g$ replacing $p = (\cdots, p_i^m, \cdots)$ by $(\cdots, g^{m\ell} / \hat{g}^\ell, \cdots)$. The aggregated CMNL model (removing the constant term $-(1/\lambda) \sum_{\ell} \hat{g}^\ell \ln \hat{g}^\ell$ of the objective function) can be stated as:

$$
\text{minimize} \quad \sum_{m, \ell} \left[ \frac{1}{\lambda} g^{m\ell} (\ln g^{m\ell} - 1) - g^{m\ell} V^{m\ell} \right], \\
\text{subject to:} \quad \sum_{m} g^{m\ell} = \hat{g}^\ell; \quad \forall (\Phi_\ell) \\
\hat{h}_r(g) \leq b_r; \quad \forall (\eta_\ell) 
$$

where $t(g) = (\cdots, g^{m\ell} / \hat{g}^\ell, \cdots)$ and $\hat{h}_r(g) = h_r(t(g)) = h_r(p)$.

A large number of exogenous system constraints may be expressed by the following linear expression:

$$
\hat{h}_r(g) = \sum_{\ell, m} a_m^\ell g^{m\ell} \leq b_r 
$$

where $a_m^\ell$ are exogenous parameters that define the amount of the scarce resource $r$ used if alternative $m$ is chosen by a user and $b_r$ is the bound for
the $r$th system constraint. Eq. (5.13) define this particular case:

$$p_{m'} = \frac{\exp\{\lambda(V_{m'} + \sum_r \eta_r a_{m'}^r)\}}{\sum_m \exp\{\lambda(V_{m} + \sum_r \eta_r a_{m}^r)\}}$$

(5.22)

Eq. (5.22) shows that users of type $\ell$ who select alternative $m'$ must pay an additional price $\sum_r \eta_r a_{m'}^r$ due to the limitation of the resources (active constraints). This price is the same for all users and produces the choice of an alternative considering the utility of other alternatives. The final price $\eta_r$ of each resource $r$ is fixed in an equilibrium situation in which no user can improve their utility by changing their selected alternative unilaterally.

### 5.2.4 The constrained hierarchical multinomial logit

In this section the constrained logit model is extended to the constrained nested logit model. A selection process similar to the one shown in Figure 5.1 is considered.

![Hierarchic nested logit model](image)

**Figure 5.1**: Hierarchic nested logit model

minimize

$$\sum_{m, \ell} \left[ \eta_1^{m\ell} g^{m\ell} (\ln g^{m\ell} - 1) + \eta_2^{m\ell} \sum_{s \in S_m} g_s^{m\ell} (\ln g_s^{m\ell} - 1) \right]$$

subject to:

$$\sum_{m} g^{m\ell} = g_x^{\ell}; \quad \forall \ell (\Phi_{\ell})$$

$$\sum_{s \in S_m} g_s^m = g^{m\ell}; \quad \forall m, \ell (\Theta_{m\ell})$$

$$h_r(g) \leq b_r; \quad \forall r (\eta_r)$$

(5.23)
where \( g = (\cdots, g_{m\ell}^t, \cdots), \eta_1^m = \frac{1}{\lambda_1}, \eta_2^m = \frac{1}{\lambda_2}, \) and \( \eta_2^m = \frac{1}{\lambda_2}. \)

Applying the KKT optimality conditions to CMNL problem:

\[
\frac{\partial L}{\partial g_{m\ell}^t} = \frac{1}{\lambda_2^m} \ln g_{m\ell}^t - V_{m\ell}^t - W_{m\ell}^t - \Theta_{m\ell} = 0 \quad (5.24)
\]

\[
\frac{\partial L}{\partial g_{m}^t} = \eta_1^m \ln g_{m}^t + \Theta_{m\ell} - \Phi_{\ell} = 0 \quad (5.25)
\]

\[\eta_r \leq 0 \quad \text{and} \quad \eta_r \left( \tilde{h}_r(g) - b_r \right) = 0 \quad (5.26)\]

where

\[W_{s}^m = \sum_r \eta_r \frac{\partial h_r}{\partial g_{m}^t}. \quad (5.27)\]

Solving for \( g_{m\ell}^t \) in (5.24),

\[
g_{s}^m = \exp \left\{ \lambda_2^m (V_{s}^m + W_{s}^m + \Theta_{m\ell}) \right\} = \exp(\lambda_2^m \Theta_{m\ell}) \exp \{ \lambda_2^m (V_{s}^m + W_{s}^m) \} \quad (5.28)\]

and summing over \( s \in S_{m'}, \) we get

\[
g^m = \sum_s g_{s}^m = \exp(\lambda_2^m \Theta_{m\ell}) \sum_s \exp \{ \lambda_2^m (V_{s}^m + W_{s}^m) \} \quad (5.29)\]

Dividing (5.28) by (5.29) we get the probabilities at lower level

\[
\frac{g_{s}^m}{g^m} = \exp \{ \lambda_2^m (V_{s}^m + W_{s}^m) \} \sum_s \exp(\lambda_2^m (V_{s}^m + W_{s}^m)) \quad (5.30)\]

Finding \( \Theta_{m\ell} \) from (5.29)

\[
\Theta_{m\ell} = \frac{1}{\lambda_2^m} \ln g^m - \frac{1}{\lambda_2^m} \ln \left[ \sum_s \exp \{ \lambda_2^m (V_{s}^m + W_{s}^m) \} \right] \quad (5.31)\]
and replacing in (5.25)

\[ \eta_1^{m\ell} \ln g^{m\ell} + \frac{1}{\lambda_2^{m\ell}} \ln g^{m'\ell} - \frac{1}{\lambda_2^{m'\ell}} \ln \left[ \sum_s \exp\{\lambda_2^{m'\ell} (V_{s}^{m'\ell} + W_{s}^{m'\ell})\} \right] - \Phi_\ell = 0 \]

\[ \frac{1}{\lambda_1^{\ell}} \ln g^{m'\ell} - L^{m'\ell} - \Phi_\ell = 0 \]  
(5.32)

where \( L^{m'\ell} \) is the classical log-sum given by

\[ L^{m'\ell} = \frac{1}{\lambda_2^{m'\ell}} \ln \left[ \sum_s \exp\{\lambda_2^{m'\ell} (V_{s}^{m'\ell} + W_{s}^{m'\ell})\} \right] \]  
(5.33)

Finding \( g^{m'\ell} \) from (5.32)

\[ g^{m'\ell} = \exp\{\lambda_1^{\ell} (L^{m'\ell} + \Phi_\ell)\} \]  
(5.34)

and adding with respect to \( m \)

\[ g^{\ell} = \sum_m \exp\{\lambda_1^{\ell} (L^{m'\ell} + \Phi_\ell)\} \]  
(5.35)

Finally, the probability of selecting alternative \( m' \) in the upper level is:

\[ \frac{g^{m'\ell}}{g^{\ell}} = \frac{\exp\{\lambda_1^{\ell} (L^{m'\ell} + \Phi_\ell)\}}{\sum_m \exp\{\lambda_1^{\ell} (L^{m'\ell} + \Phi_\ell)\}} = \frac{\exp\{\lambda_1^{\ell} L^{m'\ell}\}}{\sum_m \exp\{\lambda_1^{\ell} L^{m'\ell}\}} \]  
(5.36)

Equations (5.30) and (5.36) show that the CMNL has the same behaviour as the hierarchic logit model, but the constraints introduce a penalization in the utilities of the lower level.

### 5.3 Model behaviour and utility

In this section, the utility function is formulated in the context of dynamic choices, such as departure time choice. We assume that the systematic utility function \( V^\ell : X \subset \mathbb{R}^p \rightarrow \mathbb{R} \) belongs to a given Kernel Hilbert Space (RKHS). For simplicity only one type of user \( \mathcal{L} \) is considered, thus the index \( \ell \) is eliminated. In the case of an aggregated model being used, only the attributes \( x_m \) of some alternatives can be observed, knowing utilities \( V^{m} \). This assumption might not be realistic but the next section shows how it is calculated. In this case the objective is to estimate the function \( V(x) \)
5.3. Model behaviour and utility

such that \( V(x_m) \approx V^m \). In the disaggregated model, if attributes \( x_j \) of the individuals can be observed and attributes \( V^m_i \) are known, the objective is to estimate the \( V^m(x) \) functions for each alternative \( m \) capable of reproducing known utilities, that is \( V^m(x_i) \approx V^m_i \).

Both cases follow the same process, but the first case tries to estimate one function per type of user and the other model as many as there are alternatives \( m \). To analyse in a general form subindex \( j \) is introduced to refer to an alternative \( m \) or a user \( i \). Value \( V(x_j) \) represents the systematic utility, and \( U_j \) denotes an estimation.

5.3.1 Utility formulation

We begin with a brief review of Reproducing Kernel Hilbert Spaces and the Tikhonov Regularization Theory in RKHS. The General Theory of The Tikhonov Regularization is explained in the book of Tikhonov and Arsenin (1997) and the General Theory of RKHS is defined in Aroszajn (1950). This review has been taken from Lopez-García et al. (2014b).

**Definición 5.3.1** (Reproducing Kernel). Let \( \mathcal{H} \) be a real Hilbert space of functions defined in \( X \subset \mathbb{R}^p \) with inner product \( \langle \cdot, \cdot \rangle_{\mathcal{H}} \). A function \( K : X \times X \mapsto \mathbb{R} \) is called a Reproducing Kernel of \( \mathcal{H} \) if:

1. \( K(\cdot, x) \in \mathcal{H} \) for all \( x \in X \).
2. \( f(x) = \langle f, K(\cdot, x) \rangle_{\mathcal{H}} \) for all \( f \in \mathcal{H} \) and for all \( x \in X \).

We define the norm by \( \|f\|_{\mathcal{H}} = \langle f, f \rangle_{\mathcal{H}}^{1/2} \)

A Hilbert space of functions that admits a Reproducing Kernel is called a *Reproducing Kernel Hilbert Space* (RKHS). The reproducing Kernel of a RKHS is uniquely determined. Conversely, if \( K \) is a positive definite and symmetric kernel (Mercer kernel), then it generates a unique RKHS in which the given kernel acts like a Reproducing Kernel.

Now we briefly describe Tikhonov discrete regularization theory by RKHS for the problem at hand. Let \( K \) be a Mercer kernel and \( \mathcal{H}_K \) its associated RKHS. Consider a compact subset \( X \subset \mathbb{R}^p \) and let \( \nu \) be a Borel probability measure in \( X \times \mathbb{R} \). Let the Regression function

\[
V(x) = \int_{\mathbb{R}} y d\nu(y \mid x)
\]  

(5.37)

where \( d\nu(y \mid x) \) is the conditional probability measured on \( \mathbb{R} \). Both \( \nu \) and \( V(x) \) are unknown and what we want is to reconstruct this function.
CHAPTER 5. CONSTRAINED LOGIT MODEL: FORMULATION AND CALIBRATION

Table 5.1: Examples of kernel functions

<table>
<thead>
<tr>
<th>Name</th>
<th>Form of $K(x_i, x_j)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gaussian</td>
<td>$e^{-a|x_i - x_j|^2}$ where $|\cdot|$ is the euclidean norm $a \in \mathbb{R}^+$</td>
</tr>
<tr>
<td>Laplacian</td>
<td>$e^{-a|x_i - x_j|_1}$ where $|\cdot|_1$ is the norm 1 and $a \in \mathbb{R}^+$</td>
</tr>
<tr>
<td>Polynomial</td>
<td>$(1 + ax_i^T \cdot x_j)^b, a &gt; 0$ and $b \in \mathbb{N}$</td>
</tr>
<tr>
<td>Sigmoidal</td>
<td>$\tanh(a(x_i^T \cdot x_j) + b), a \in \mathbb{R}^+, b \in \mathbb{R}$</td>
</tr>
<tr>
<td>Multiquadratic</td>
<td>$\sqrt{</td>
</tr>
<tr>
<td>Inverse multiquadratic</td>
<td>$\sqrt{</td>
</tr>
</tbody>
</table>

Let $X_n := \{x_1, \ldots, x_n\} \subset X$ and let $W_n$ be a random sample independently drawn from $\nu$ and $V(x)$ on $X$. That is:

$$W_n := \{(x_j, U_j) \in X \times \mathbb{R}\}_{j=1}^n.$$

Tikhonov Regularization Theory considers the function space

$$\mathcal{V}_n := \text{span} \{ K(\cdot, x) : x \in X_n \} \quad (5.38)$$

where $\text{span}$ is the linear hull and projects $V(x)$ onto this space by using the sample $W_n$. Tikhonov Regularization Theory makes a stable reconstruction of $V(x)$ by solving the following optimization problem:

$$V^* := \arg \min_{V \in \mathcal{V}_n} \frac{1}{n} \sum_{j=1}^n (V(x_j) - U_j)^2 + \gamma \|V\|^2_{\mathcal{H}_K} \quad (5.39)$$

where $\gamma > 0$ and $\|V\|_{\mathcal{H}_K}$ represents the norm of $V$ in $\mathcal{H}_K$. The solution $V^*$ of (5.39) is called the Regularized $\gamma$-Projection of $V(x)$ onto $\mathcal{H}_K$ associated to the sample $W_n$.

The representation theorem gives a closed form solution of $V^*(x)$ for the optimization problem (5.39). This theorem was introduced by Kimeldorf and Wahba (1970) in a spline smoothing context and has been extended and generalized to the problem of minimizing risk of functions in RKHS, see Schölkopf et al. (2001) and Cox and O’Sullivan (1990)

**Theorem 5.3.1** (Representation). Let $W_n$ be a sample of $V(x)$, let $K$ be a (Mercer) kernel and let $\gamma > 0$. Then there is a unique solution $V^*$ of (5.39) that admits a representation by

$$V^*(x) = \sum_{j=1}^n \alpha_j K(x, x_j), \text{ for all } x \in X,$$  

(5.40)
where $\alpha = (\alpha_1, \cdots, \alpha_n)^T$ is a solution to the linear equation systems:

$$\gamma n I_n + K_x \alpha = U, \tag{5.41}$$

where $I_n$ is the identity matrix $n \times n$, $U = (U_1, \cdots, U_n)^T$ and the matrix $K_x$ is given by $(K_x)_{ij} = K(x_j, x_j)$. The expression (5.40) leads to the estimate of $V(x)$ in $X_n$:

$$\hat{V}^* = K_x \alpha \tag{5.42}$$

5.4 Calibration issues

Linear utilities among the attributes is the most commonly-used approach in literature. In this Chapter the temporal nature of attributes is considered, so non-linear utilities like the kernel utilities defined in previous section appear to be better suited to the problem.

Assume a sample of $N$ decision-makers, $N_\ell$ is the number of individuals of type $\ell$. Also suppose that the number of individuals of type $\ell$ who select alternative $m$ is denoted by $N_{m,\ell}$ and it is known for a set of combinations $(m,\ell) \in D_0$. Denote $N = (\cdots, N_{m,\ell}, \cdots)$ with $(m,\ell) \in D_0$.

Assume that the vector of attributes for each alternative $m$ is known and is denoted by $x_m$. To the author’s knowledge, and in contrast to the approach in the literature, the estimated parameters are a subset of utilities instead of the attribute weightings. Let $D_1$ be the alternative subset $(m,\ell)$ in which the utility will be estimated, and let $D_2$ be the alternatives in which the utility will be calculated from the estimated function $V^\ell(x)$. The set of alternatives is decomposed in $D = D_1 \cup D_2$ with $D_1 \cap D_2 = \{\emptyset\}$.

As a first step, the vector of utilities is estimated

$$\hat{V}_1 = (\cdots, V^{m\ell}, \cdots); \ (m,\ell) \in D_1 \tag{5.43}$$

and in the second stage the utility function $V^\ell(\cdot)$ is calculated using Eq. (5.40)-(5.41) and all-non estimated utilities $V^{m\ell} = V^\ell(x_m)$ are completed. Denote

$$\hat{V}_2 = (\cdots, V^\ell(x_m), \cdots); \ (m,\ell) \in D_2 \tag{5.44}$$

The above two stages are schematically represented by

$$\hat{V}_2 = \mathcal{H}(\hat{V}_1) \tag{5.45}$$
CHAPTER 5. CONSTRAINED LOGIT MODEL: FORMULATION AND CALIBRATION

CMNL parameters are $V, \lambda, b_r, \hat{g}^r$. It is assumed that the upper bounds $b_r$ of the system constraints and $\hat{g}^r$ are known. The remaining parameters $(V, \lambda)$ must be calibrated. In this section a generic calibration methodology is described without specifying if it is a hierarchic model or not, removing the related indices of the parameters. As CMNL is a strictly convex program it poses a single optimum and CMNL defines implicitly a function, which obtains the disaggregation of the demand by alternatives for each pair $(V, \lambda)$:

$$g = \text{CMNL}(V, \lambda)$$  (5.46)

Using Eq. (5.45),

$$g = \text{CMNL}((\hat{V}_1, \mathcal{H}(\hat{V}_1)), \lambda)$$  (5.47)

Finally the calibration problem can be stated as:

$$\begin{align*}
\text{minimize} & \quad (\hat{V}_1, \lambda) \quad F(g, N), \\
\text{subject to} & \quad g = \text{CMNL}((\hat{V}_1, \mathcal{H}(\hat{V}_1)), \lambda)
\end{align*}$$  (5.48)

where $F$ is a similarity function between predicted demand by CMNL model, $g$, and the observed values, $N$.

A maximum likelihood (ML) approach:

$$F(g, N) = \ln L = \sum_{(m, \ell) \in D_0} N_{m, \ell} \ln p^{m\ell} = \sum_{(m, \ell) \in D_0} N_{m, \ell} \ln \left(\frac{g^{m\ell}}{N_{\ell}}\right)$$  (5.49)

leading to the optimization problem

$$\begin{align*}
\text{max} & \quad \sum_{(V, \lambda) \in D_0} N_{m, \ell} \ln \left(\frac{g^{m\ell}}{N_{\ell}}\right) \\
g & = \text{CMNL}((\hat{V}_1, \mathcal{H}(\hat{V}_1)), \lambda)
\end{align*}$$  (5.50)

In some cases, like in the numerical experiments of this work, disaggregated values by alternatives $N_{m, \ell}$ are not known. In these cases the least squares method can be used:

$$F(g, N) = \sum_{(m, \ell) \in D_0} \gamma^{m\ell} (g^{m\ell} - N_{m, \ell})^2$$  (5.51)

where $\gamma^{m\ell}$ are weightings.
5.4. Calibration issues

The likelihood maximization or generalized least squares technique is achieved by embedding the computation of $F$ within a non-linear optimization framework as shown in Figure 5.2.

**Figure 5.2:** Schematic representation of CMNL calibration procedure

5.4.1 Overspecification

In this subsection we show that there exist infinite solutions for the calibration problem (5.48). This is due to over-specification of the parameters (García-Ródenas and Marín (2009), Bierlaire et al. (1997), Daganzo and Kusnic (1993)). Parameter over-specification must be avoided because although some of the more robust methods succeed in solving the problem, their speed of convergence may be very slow. This problem is due to the singularity of the second derivative matrix of the log-likelihood function.

5.4.1.1 First source of over-specification

The first source of over-specification arises in the interaction between the structure of the utilities and the parameters $\lambda_j^\ell$ with $j \in \{1, 2\}$ and it becomes:

$$
\begin{align*}
\tilde{V}_m^\ell & = \delta^\ell \tilde{V}_m^\ell \\
\tilde{\lambda}_j^\ell & = \frac{\lambda_j^\ell}{\delta^\ell}; \quad j \in \{1, 2\}
\end{align*}
$$

$\ell \in \mathcal{L}$
The above relationships are schematically denoted as \( \tilde{V} = \delta \hat{V} \) and \( \tilde{\lambda} = \lambda / \delta \).

Let \((\hat{V}_1, \lambda)\) be a vector of parameters for the CMNL model and let

\[ g = \text{CMNL}((\hat{V}_1, \mathcal{H}(\hat{V}_1)), \lambda) \]  

(5.52)

be the estimated demand.

The objective function of the CMNL model is separable in \( \ell \). If each term in \( \ell \) is multiplied by the constant \( \delta^\ell > 0 \) then the optimal solution associated with \( \ell \) is not changed. Moreover, the system constraints hold. This leads to:

\[ g = \text{CMNL}((\delta \hat{V}_1, \delta \mathcal{H}(\hat{V}_1)), \lambda / \delta) = \text{CMNL}((\tilde{V}_1, \tilde{\mathcal{H}}(\tilde{V}_1)), \tilde{\lambda}) \]  

(5.53)

It is worth noting that the utilities \( \hat{V}_1 \) are multiplied by \( \delta \), the solution of system (5.41) is multiplied by \( \delta \) and thus the utilities \( \tilde{V}_2 \) are also multiplied by \( \delta \) because they are linear on their parameters \( \alpha \). Mathematically

\[ \mathcal{H}(\delta \tilde{V}_1) = \delta \mathcal{H}(\tilde{V}_1) \]  

(5.54)

Using (5.52), (5.53) and (5.54), we obtain

\[ g = \text{CMNL}((\hat{V}_1, \mathcal{H}(\hat{V}_1)), \lambda) = \text{CMNL}((\tilde{V}_1, \tilde{\mathcal{H}}(\tilde{V}_1)), \tilde{\lambda}) \]  

(5.55)

As objective function of calibration model (5.48), \( F(g, N) \), depends only on \( g \), both solutions \((\hat{V}_1, \lambda)\) and \((\tilde{V}_1, \tilde{\lambda})\) have the same objective value. This fact shows that there exist infinite optimal solutions of the calibration model.

Thus the scale parameters of Gumbel error terms are undetermined. In practice, setting one Gumbel term for each \( \ell \) is sufficient for the identification.

5.4.1.2 Second source of over-specification

The second source of over-specification in the constrained hierarchical multinomial logit models is adding the same value to the utilities of all the alternatives, which does not affect the log-likelihood of the sample. In this case, we assume that \( D_2 = \{\emptyset\} \). The set of utilities

\[ \tilde{V}_s = \hat{V}_s^{m\ell} + \gamma^\ell; \ell \in \mathcal{L} \]  

(5.56)

produces the same solution as the optimization model. If utilities \( \tilde{V} \) of the objective function CMNL are replaced by utilities \( \hat{V} \) the same objective
function value plus the constant is obtained

\[- \sum_{s \in S} \tilde{V}_m^{\ell} m^\ell = - \sum_{s \in S} (\tilde{V}_s^{\ell} + \gamma^\ell) g_s^m = - \gamma^\ell \tilde{g}^\ell - \sum_{s \in S} \tilde{V}_s^{\ell} g_s^m\] (5.57)

Bierlaire et al. (1997) have analysed over-specification in nested logit models to the log-likelihood function. These authors have analysed the relationship between any two arbitrary strategies to avoid over-specification, and shown that the two strategies are equivalent under a linear transformation of the variables. Some algorithms are independent of such transformations: Newton’s method and the quasi-Newton methods of the Broyden family are combined with line searches. If these are used, then the way in which the over-specification is eliminated is not important. Daganzo and Kusnic (1993) suggested equating one parameter to zero for each set of parameters jumbled in every source of over-specification, and estimating the rest in order to avoid over-specification in the nested logit model.

### 5.5 Numerical analysis

The objective of this section is to test the viability of the calibration of the constrained logit model following the proposed approach based on non-linear utilities and free derivative methods.

#### 5.5.1 Application of the constrained hierarchical multinomial logit model for railway service modelling

In this section a constrained MNL model is proposed to solve the problem of calculating an optimal schedule in railway systems.

![Diagram](image)

**Figure 5.3:** Hierarchical MNL demand for railway service choice

Suppose there are various types of users varying by reason for journey.
economic and social characteristics of the traveller and his/her origin-destination.

In this example the type of users \( \ell \) have been disaggregated into two components \( \ell = (\ell_1, \ell_2) \). The first member includes the socio-economic characteristics and reason for travel, the second component represents the origin and destination of the trip. Each type of user is denoted as \((\ell, \omega)\). Index \( \ell \in \mathcal{L} \) refers to the first factors and \( \omega = (i, j) \in \mathcal{W} \) is a trip from station \( i \) to station \( j \).

Moreover round trip demands in the same planning period are considered, and will be explained below, after analysing \( i \rightarrow j \). Assume a potential demand \( \{\hat{g}_{\omega}^{\ell}\}_{\omega \in \mathcal{W}} \) for a demand type \( \omega \) for users \( \ell \in \mathcal{L} \). Assume the total demand disaggregated in two alternatives:

(a) (High-speed) train trips.

(b) Another mean of transport trip.

Assume a logit model which divides the potential demand between alternatives (a) and (b):

\[
g_{\omega}^{m\ell} = \frac{\exp(\lambda_1 V_{\omega}^{m\ell})}{\sum_{m \in \{a,b\}} \exp(\lambda_1 V_{\omega}^{m\ell})} \cdot \hat{g}_{\omega}^{\ell} \quad m \in \{a,b\}, \ell \in \mathcal{L} \tag{5.58}
\]

where \( V_{\omega}^{m\ell} \) is the utility of alternative \( m \) for user type \( \ell \) and demand type \( \omega \).

The model consider a nested logit model for disaggregate the demand considering the feasible timetable for a trip type \( \omega \). Denote as \( S_\omega \) the feasible set of services for making a trip type \( \omega \). The second level of the nested logit model disaggregate the demand between the different services:

\[
g_{\omega,s}^{a\ell} = \frac{\exp(\lambda_2 V_{\omega,s}^{a\ell})}{\sum_{s' \in S_\omega} \exp(\lambda_2 V_{\omega,s'}^{a\ell})} \cdot g_{\omega}^{a\ell} \quad s \in S_\omega, \ell \in \mathcal{L} \tag{5.59}
\]

Nested logit models calculate the utility of alternative \( V_{\omega}^{a\ell} \) as the “log-sum” of the utilities of each service:

\[
V_{\omega}^{a\ell} = \frac{1}{\lambda_2} \ln \left( \sum_{s \in S_\omega} \exp(\lambda_2 V_{\omega,s}^{a\ell}) \right) \tag{5.60}
\]

The above nested logit model is combined with the capacity constraints of the trains. When a train reaches station \( j \) the vehicle has picked up
5.5. Numerical analysis

passengers from preceding stations. The number of passengers that can take the train is then restricted by the capacity of the vehicle. Denote by \( W^+_s \) the set of origin-destination pairs whose users take the service \( s \) before station \( j \) and leave the vehicle after station \( j \). Also denote by \( W^+_{sj} \) the origin-destination pairs \( \omega \) whose origin is station \( j \) and use \( s \). The capacity constraints of the service \( s \) in station \( j \) is formulated as:

\[
\sum_{\omega \in W^+_{sj}} g^a_{\omega,s} + \sum_{\omega \in W^+_{sj}} g^b_{\omega,s} \leq K_s \quad \text{for all } s \in S, \quad j \in J_s; 
\]

(5.61)

where \( K_s \) is the capacity of train \( s \), \( S \) is the set of services and \( J_s \) represents the set of stations in which the service \( s \) will stop.

Eq. (5.61) can be reformulated, disaggregated by users:

\[
\sum_{\ell \in L} \left[ \sum_{\omega' \in W^+_{sj}} g^{a\ell}_{\omega',s} + \sum_{\omega \in W^+_{sj}} g^{b\ell}_{\omega,s} \right] \leq K_s \quad \text{for all } s \in S, \quad j \in J_s; 
\]

(5.62)

minimize

\[
\sum_{\omega \in W} \sum_{\ell \in L} \left[ \sum_{m \in \{a,b\}} \eta^m g^m(\ln g^m - 1) + \eta^c g^c(\ln g^c - 1) - V_{b\ell} g^b - \sum_{s \in S} V_{a\ell} g^a \right],
\]

subject to:

\[
g^a + g^b = \hat{g}^\ell, \quad \omega \in W, \ell \in L
\]

\[
g^a = \sum_{s \in S} g^{a\ell}_{\omega,s}, \omega \in W, \ell \in L
\]

\[
\sum_{\ell \in L} \left[ \sum_{\omega' \in W^+_{sj}} g^{a\ell}_{\omega',s} + \sum_{\omega \in W^+_{sj}} g^{b\ell}_{\omega,s} \right] \leq K_s, s \in S, \quad j \in J_s;
\]

(5.63)

where \( \eta^a = \frac{1}{\lambda^1_1} - \frac{1}{\lambda^1_2}, \eta^b = \frac{1}{\lambda^1_1}, \eta^c = \frac{1}{\lambda^1_2} \).

This Chapter deals with over-specification based on two considerations. Firstly, the non-observed utilities are set to \( V^+_{b\ell} = 0 \) (the second source of over-specification). The second issue considers that each interval

\[
-B \leq V^+_{a\ell} \leq B
\]

(5.64)

with \( B > 0 \) contains optimal solutions of the calibration problem (the first source of over-specification), thus the imposition of this constraint limits the search space without reducing the quality of the fit. Selecting limits with very small \( B \) value equivalent to \( \delta \rightarrow 0 \) could lead to large values of \( \lambda^i_j \), causing a more complex estimation. A trade-off between the range of the interval of the utilities and the order of magnitude of the parameters \( \lambda^i_j \) must be achieved.
5.5.2 Case study

To test the model given in Section 5.5.1, a case study has been generated. This numerical example is focused on the Madrid-Seville corridor of the High Speed Railway network of Spain. This corridor is formed by 5 stations (MAD, CR, PU, COR, SEV) which produces 20 origin-destination demand pairs (10 per direction of the travel) formed by 15115 passengers/day. Currently this demand is completely covered by 100 services. Figure 5.4 depicts the corridor used by these services, Table 5.2 indicates the route of each type of service and Table 5.3 shows the maximum capacity of each type of train.

![Figure 5.4: Madrid-Seville corridor](image)

In this example only one type of user $\ell = 1$ is considered attending to economic characteristics, but taking into account the journey of each traveller (origin-destination pair $\omega$), there will be 20 types of users. Each user type $\omega$ can travel using a set of services $s$. Considering the planned schedule, the set of alternatives $(\omega, s) \in D$ consists of 298 possibilities. In this case the proposed model could estimate 298 parameters.

To calibrate the model 25 of the services of the complete schedule have been selected randomly, generating a set $D_1$ with 66 possibilities $(\omega, s)$ and, consequently, 66 parameters $V_{\omega,s}$ that should be estimated to calculate the utility of each alternative as explained in Section 5.3. Note that the solution of the linear system (5.41) produces the values $\alpha_{\omega,s}$.

The attributes considered for each possible alternative $(\omega, s)$ are: i) the price $x_{\omega,s}^1$, ii) the travel time $x_{\omega,s}^2$ and iii) the timetable $x_{\omega,s}^3$. The vector of attributes is denoted as $x_{\omega s} = (x_{\omega,s}^1, x_{\omega,s}^2, x_{\omega,s}^3)$. The data used for the experiment can be downloaded from the links in Appendix B.

To simplify we have set $V^\omega(x) = V^s(x)$ for all pairs $\omega$ in this experiment,
5.5. Numerical analysis

<table>
<thead>
<tr>
<th>Type of service</th>
<th>Route</th>
<th>Amount of services</th>
<th>Type of train</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>MAD → CR → PU</td>
<td>11</td>
<td>AVANT</td>
</tr>
<tr>
<td>2</td>
<td>MAD → CR → PU → COR</td>
<td>3</td>
<td>AVE</td>
</tr>
<tr>
<td>3</td>
<td>MAD → CR → PU → COR → SEV</td>
<td>5</td>
<td>AVE</td>
</tr>
<tr>
<td>4</td>
<td>MAD → COR</td>
<td>4</td>
<td>AVE</td>
</tr>
<tr>
<td>5</td>
<td>MAD → COR → SEV</td>
<td>9</td>
<td>AVE</td>
</tr>
<tr>
<td>6</td>
<td>MAD → SEV</td>
<td>3</td>
<td>AVE</td>
</tr>
<tr>
<td>7</td>
<td>COR → SEV</td>
<td>6</td>
<td>MD</td>
</tr>
<tr>
<td>8</td>
<td>COR → SEV</td>
<td>9</td>
<td>AVANT</td>
</tr>
<tr>
<td>9</td>
<td>SEV → COR</td>
<td>6</td>
<td>MD</td>
</tr>
<tr>
<td>10</td>
<td>SEV → COR</td>
<td>9</td>
<td>AVANT</td>
</tr>
<tr>
<td>11</td>
<td>SEV → COR → PU → CR → MAD</td>
<td>5</td>
<td>AVE</td>
</tr>
<tr>
<td>12</td>
<td>SEV → COR → MAD</td>
<td>8</td>
<td>AVE</td>
</tr>
<tr>
<td>13</td>
<td>SEV → MAD</td>
<td>3</td>
<td>AVE</td>
</tr>
<tr>
<td>14</td>
<td>COR → PU → CR → MAD</td>
<td>3</td>
<td>AVE</td>
</tr>
<tr>
<td>15</td>
<td>COR → MAD</td>
<td>5</td>
<td>AVE</td>
</tr>
<tr>
<td>16</td>
<td>PU → MAD → CR</td>
<td>11</td>
<td>AVANT</td>
</tr>
</tbody>
</table>

Table 5.2: Types of railway services on Madrid-Seville corridor

<table>
<thead>
<tr>
<th>Type</th>
<th>Train capacity</th>
</tr>
</thead>
<tbody>
<tr>
<td>AVE</td>
<td>308 (passengers)</td>
</tr>
<tr>
<td>AVANT</td>
<td>237</td>
</tr>
<tr>
<td>MD</td>
<td>190</td>
</tr>
</tbody>
</table>

Table 5.3: Capacity of each type of train on Madrid-Seville corridor

and the utility function $V^*(x)$ is defined as:

$$V^*(x) = \sum_{(\omega,s) \in D_1} \alpha_{\omega,s} K(x,x_{\omega,s}), \text{ for all } x \in X,$$

(5.65)

where a Gaussian kernel $K(x,y) = e^{-a\|x-y\|^2}$ is used in which $\| \cdot \|$ is the Euclidean norm $a \in R^+$. In this case $a = 5$ has been considered. The regularization parameter $\gamma$ has been defined as 0.00001.

5.5.3 Estimation methods

The data used as $N$ are public domain and therefore show aggregated information as the total demand in a determined origin-destination pair or for a type of service. A calibration procedure based on a ME approach is not available because the disaggregated observations for each pair $(\omega,s)$ are unknown. In this test we have adapted the generalized least squares
technique for comparing the known demand behaviour versus the demand predicted by the CMNL model. In this example the values of $\lambda_1$ and $\lambda_2$ have been also calibrated.

The optimization method selected for solving the calibration problem (5.48) is a hybridization of the Standard Particle Swarm Optimization (SPSO) (Zambrano-Bigiarini et al. (2013)) and the Nelder-Mead (NM) (Nelder and Mead (1965)) algorithm based on the framework presented in Espinosa-Aranda et al. (2013b). Hybrid algorithms try to make full use of merits of various optimization techniques in order to obtain an efficient method in the search for global optima.

The SPSO has been used successfully in global optimization problems particularly in transportation research (see Angulo et al. (2011) and Angulo et al. (2013)). The main advantages of PSO algorithms could be summarized as follows: they are capable of avoiding local optima, doing a search in the entire solution space, also are robust against initialization parameters, viable, efficient with a smaller computational burden and have a simple selection of the right parameter values. The NM method is a direct search method that does not use numerical or analytic gradients and has local convergence with a high exploitation capacity.

The main issue analysed in this section is the applicability of the proposed methodology, because the calibration problem to be solved has a bi-level nature in which, to evaluate the objective function, an equilibrium model must be solved. Moreover, a linear equation system for estimating $\alpha$ (see Eq. 5.41) is solved in each iteration and there is no information about the derivatives because of the implicit nature of the function. All of this results in a high computational burden.

The resolution procedure for the CMNL model has been GAMS+CONOPT 24, showing that in an Intel I7 4 Cores 3.2 GHz with 16 GB RAM computer the CPU time for each problem is around 0.12 seconds. This reduced computational burden the coding of a Lagrangian Relaxation algorithm unnecessary.

The SPSO+NM has been implemented in MATLAB, which calls GAMS to solve each individual CMNL problem. The stopping criterion is based on the total number of solved CMNL models. The SPSO algorithm was run for 50,000 objective function evaluations. A random start on an interval defined by Eq. (5.64) was used. The size of the swarm was 40 particles. The PSO-parameters $w, c_1$ and $c_2$ for updating velocity are defined as $w = 1/(2 \ln(2))$, $c_1 = 0.5 + \ln(2)$ and $c_2 = c_1$ (Zambrano-Bigiarini et al. (2013)). NM is run for 50,000 function evaluations starting from the best solution found by the SPSO algorithm to improve the solution.

Table 5.4 shows the computational results depending on the feasible region considered (5.64). The mean computational cost of each run of the SPSO algorithm was 1.7 hours, and with NM, 4.4 hours. Therefore the calibration of the CMNL model can be computed in an affordable time.
5.5. Numerical analysis

<table>
<thead>
<tr>
<th>Case</th>
<th>-B</th>
<th>B</th>
<th>SPSO</th>
<th>SPSO+NM</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-0.5</td>
<td>0.5</td>
<td>4.7276E+05</td>
<td>2.5866E+05</td>
</tr>
<tr>
<td>2</td>
<td>-1</td>
<td>1</td>
<td>3.2272E+05</td>
<td>2.5248E+05</td>
</tr>
<tr>
<td>3</td>
<td>-2</td>
<td>2</td>
<td>3.2828E+05</td>
<td>2.5988E+05</td>
</tr>
<tr>
<td>4</td>
<td>-5</td>
<td>5</td>
<td>4.8158E+05</td>
<td>2.7113E+05</td>
</tr>
<tr>
<td>5</td>
<td>-10</td>
<td>10</td>
<td>5.6018E+05</td>
<td>2.6979E+05</td>
</tr>
<tr>
<td>6</td>
<td>-30</td>
<td>30</td>
<td>1.1273E+06</td>
<td>1.1273E+06</td>
</tr>
<tr>
<td>7</td>
<td>-50</td>
<td>50</td>
<td>1.2238E+06</td>
<td>1.2238E+06</td>
</tr>
<tr>
<td>8</td>
<td>-100</td>
<td>100</td>
<td>1.4023E+06</td>
<td>1.4023E+06</td>
</tr>
<tr>
<td>9</td>
<td>-500</td>
<td>500</td>
<td>1.1742E+06</td>
<td>1.1742E+06</td>
</tr>
<tr>
<td>10</td>
<td>-1000</td>
<td>1000</td>
<td>1.0889E+06</td>
<td>1.0628E+06</td>
</tr>
<tr>
<td>11</td>
<td>-5E+03</td>
<td>5E+03</td>
<td>1.2428E+06</td>
<td>1.2428E+06</td>
</tr>
<tr>
<td>12</td>
<td>-1E+04</td>
<td>-1E+04</td>
<td>1.0640E+06</td>
<td>1.0640E+06</td>
</tr>
<tr>
<td>13</td>
<td>-5E+04</td>
<td>5E+04</td>
<td>1.3045E+06</td>
<td>1.3045E+06</td>
</tr>
<tr>
<td>14</td>
<td>-5E+05</td>
<td>5E+05</td>
<td>1.2318E+06</td>
<td>1.2318E+06</td>
</tr>
<tr>
<td>15</td>
<td>-1E+07</td>
<td>1E+07</td>
<td>2.2174E+06</td>
<td>2.2113E+06</td>
</tr>
</tbody>
</table>

Table 5.4: Solution found versus amplitude of feasible region

As can be seen, the results show that with small intervals the algorithms are capable of finding a better solution than when searching in a bigger feasible region. This can also be seen in Figure 5.5 which depicts the evolution of the objective function during the running of the SPSO algorithm per case study. The red graphs represent case studies 1-5, the blue 6-14 and the black line 15.

5.5.4 Study of the best solution obtained

The best solution is obtained by using the interval ($-1, 1$) as the space of parameters. This section shows this solution. Figures 5.6 and 5.7 depict the utility function fixing respectively the departure time at 8:30 and the travel time to 63 minutes.

It can be seen in Figure 5.6 that the specification of $V^*(x) = V^*(x)$ for each pair $\omega$ produces the travel time attribute with which to consider the demand effect in each pair $\omega$. For example, the largest travel time represents the largest trip $\omega = (MAD, SEV)$ while the smallest time occur in pair $\omega = (CR, PU)$. The utility function captures the demand in each pair.

Figure 5.7 depicts the results obtained for a fixed travel time of 63 minutes (i.e. fixed a o-d pair $\omega$). This case shows how the utility change depending on the departure time coinciding with the demand peaks at determined hours (8:30, 15:00 and 19:30).

Finally, note that an analytical expression of $V^*(x)$ is computed and it is possible to calculate the marginal utilities $\frac{\partial V^*}{\partial x_{\omega,i}}$ for $i = 1, 2, 3$. 
CHAPTER 5. CONSTRAINED LOGIT MODEL: FORMULATION AND CALIBRATION

5.6 Conclusions

In this Chapter, a constrained MNL formulation is presented to model both the dynamic and constrained decision spaces in discrete choice contexts. The estimation procedure and specification issues associated with the CMNL formulation are also discussed.

This Chapter presents a new method for the calibration of wide transport-related choice phenomena based on the consideration of utilities as parameters for the estimation instead of classical attribute weights and the use of Kernel Hilbert Spaces for the specifications of utility functions.

A novel railway demand model is used to test the suitability of the proposed approach. A computational experiment based on real data for the Madrid-Seville high-speed corridor is studied. The results show how the CMNL model could represent the behaviour of the users of the railway network.

Figure 5.5: Evolution of the SPSO algorithm for all cases
5.6. Conclusions

Figure 5.6: Utility estimation with fixed departure time (8:30)

Figure 5.7: Utility estimation with fixed travel time (63 min)
6. High-Speed railway scheduling
This Chapter proposes an optimization model for High-Speed railway scheduling. The model is composed of two sub-models. The first is a discrete event simulation model which represents the supply of the railway services whereas the second is a constrained logit-type choice model which takes into account the behaviour of users. This discrete choice model evaluates the attributes of railway services such as the timetable, price, travel time and seat availability (capacity constraints) and computes the High-Speed railway demand for each planned train service.

A hybridization of the Standard Particle Swarm Optimization and Nelder-Mead methods has been applied for solving the proposed model and a real case study of the High-Speed corridor Madrid-Seville in Spain has been analysed. Furthermore, parallel computation strategies are used to speed up the proposed approach.

6.1 Introduction

Currently High-Speed Railway (HSR) systems are expanding and incrementing their demand share. Analytical tools for the timetable setting problem are essential to increase the competitiveness of the rail industry. Therefore Train Timetabling Problem (TTP) has been addressed widely in the literature in the last thirty years. High-speed systems are an emergent technology whose singular features have not been dealt with in the current literature on solving the TTP. The travel time, the ticket price and the competition of these systems with the traditional air, rail and private vehicles should be taken into account. This shows the necessity in this type of TTP models of knowing the behaviour of the passengers with respect to the attributes of the proposed timetable, thus, High-Speed demand forecasting is required. Cascetta and Coppola (2013) consider that a High-Speed demand approach consists
of three main components:

1. The diverted demand, which represents the choice of a passenger between other means of transport (plane, car, other rail services, etc) and HSR.

2. The induced demand, which depends directly on the characteristics of the HSR services offered (ticket cost, travel time, timetable, etc), or indirectly due to modifications of the travellers’ mobility or lifestyle choices.

3. The economy-based demand growth, which is linked to the trends of the economic system, considering that people travel more when they are wealthy.

Diverted demand and directly induced demand are considered as endogenous factors of the HSR system. Indirectly induced demand and economy-based demand growth are considered as exogenous factors of the HSR system. A model which tries to solve the TTP for HSR systems must be capable of modelling the endogenous factors of the system.

HSR demand forecasting models can also be classified as aggregate or disaggregate. Aggregate models are based on aggregate demand elasticity values and make use of large data sets obtained from ticket sales and surveys. They are useful for rail demand growth predictions (see Wardman (2006)) but are limited when big changes happen. Another limitation of these models is that they cannot simulate flows on individual rail segments or trains.

Disaggregate models consider the individual as the basic unit of observation, dividing the passengers into different types depending on different factors. These models are consistent with travel choice theory using data at the level of individual travellers.

Disaggregate models can be classified as mono-modal or multi-modal. Mono-modal models are focused on a specific transport mode and forecast the demand based on the operational improvement of this mode (for some examples see Couto and Graham (2008) and Hsu and Chung (1997)). These models are typically conceived to represent the behaviour of the passengers in an urban context.

Multi-modal models include the competition between HSR and other modes of transports eligible for the same trip. Most of these models focus on the competition between HSR and air transportation (see Park and Ha (2006), Román et al. (2007) and Fröidh (2008)), some include cars (see Mandel et al. (1997), Yao and Morikawa (2005) and Martin and Nombela (2007)) and very few add the competition between HSR operators (see Ben-Akiva et al. (2010) and Cascetta and Coppola (2012)).
6.1. Introduction

Furthermore the modelling approach of demand forecasting models can be divided into a frequency-based approach and a schedule-based approach. The frequency-based approach is focused on defining a frequency for each type of service over given time intervals, constraining the modelling possibilities in low-frequency systems or operational planning. In such cases, scheduled services can be represented as individual trips following a schedule-based approach defining a timetable (see Wilson and Nuzzolo (2004)).

The majority of the models presented in the literature to forecast HSR demand follow a frequency-based demand approach. Schedule-based studies like the defined in Chapter 5 where a constrained nested logit model is proposed with the objective of predicting the demand in a HSR system are scarce. This model considers the timetable, price of the ticket, travel time and capacity of the trains in the selection process. These authors use Reproducing Kernel Hilbert spaces to consider dynamic utilities, which is an essential issue for modelling the departure time choice.

Table 6.1 shows a taxonomy of the models of the current state of the art.

<table>
<thead>
<tr>
<th>Frequency-based</th>
<th>Schedule-based</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multi-modal</td>
<td></td>
</tr>
<tr>
<td>Single rail service</td>
<td>Roman et al. (2007), Yao and Morikawa (2005), Fröidh (2008)</td>
</tr>
<tr>
<td>Mono-modal</td>
<td></td>
</tr>
<tr>
<td>Multi rail service</td>
<td>Couto and Graham (2008), Hsu and Chung (1997)</td>
</tr>
<tr>
<td>Single rail service</td>
<td>Urban case</td>
</tr>
</tbody>
</table>

Table 6.1: Review of HSR disaggregate models (source: Cascetta and Coppola (2013))

The other main issue in TTP is the modelling of the railway services supplied. Traditionally to represent this problem several types of constraints like block capacity constraints, flow constraints, priority constraints, train service capacity constraints and rolling stock constraints have been considered (see Cordeau et al. (1998)). HSR systems have received a great deal of investment which eases the operational control of these systems allowing the simplification of the traditional regional or intercity railway systems modelling.

The proposed HSR-TTP approach can be classified as a disaggregate multi-modal schedule-based model and focuses on the endogenous factors associated with HS passenger behaviour. This model has the availability of
being applied to a complete HSR network and the aim is the benefit of the HSR operator.

This Chapter is organized as follows. Section 6.2 defines the proposed HSR-TTP model, Section 6.3 explains the algorithms used for solving this model, in Section 6.4 several computational experiments are reported, and finally Section 6.5 concludes with a discussion of our findings and future work.

6.2 A model for HSR scheduling

This section sets out a TTP model focused on HSR operator’s profit. This model, whose structure is depicted in Figure 6.1, optimizes the timetable, taking into account the behaviour of the passengers.

This HSR-TTP model comprises three modules:

1. **Supply model.** This model represents the movements of trains on the HS railway network.
2. A model for computing the *utilities* of the scheduled services as a function of their attributes.
3. **HSR demand model.** This model defines the choices of the users on the HSR network.

![Figure 6.1: HSR-TTP timetable model based on user behaviour](image)

For convenience, the list of variables, sets and indices used is shown in Table 6.2.

6.2.1 Supply model

The set of constraints defined in this section model railway operations and overtaking between trains. A discrete event model is formulated to obtain
6.2. A model for HSR scheduling

Sets and indices
- \( s \): Railway service. It is the route of a train between two end stations.
- \( S \): Set of services of the railway network.
- \( i, j \): Stations.
- \( J \): Set of stations of the railway network.
- \( k, k' \): Trains.
- \( K \): Set of trains on the railway network.
- \( w \): Origin-destination demand pair.
- \( W \): Set of origin-destination demand pairs.
- \( S_w \): Set of services which could cover demand \( w \).
- \( E \): Set of events of the railway network.

Variables
- \( e \): Event in the railway network.
- \( T_s \): Departure time of service \( s \) at the initial station.
- \( T_{aj}^s \): Instant in which a service \( s \) arrives at station \( j \).
- \( T_{dj}^s \): Instant in which a service \( s \) departs from station \( j \).
- \( \delta_e \): Instant associated with event \( e \).
- \( \tau_{aj}^s \): Last train, with respect to the current simulation clock, arrives at station \( j \).
- \( \tau_{dj}^s \): Last programmed departure instant from station \( j \) before the arrival of the next service at station \( j \) with respect to the current simulation clock.
- \( d_{ij} \): Dwell time of railway track segment \( i \to j \).
- \( c_{kj}^i \): Travel time of the train \( k \) in the track segment \( i \to j \).
- \( p_{s_j} \): Stop time of service \( s \) in station \( j \).
- \( \tilde{T}_s' \): Real instant when service \( s' \) starts.
- \( T_s' \): Planned initial time of service \( s' \).
- \( \delta_s \): End time of service \( s \).
- \( r_{ss'} \): Rolling stock time for getting the train ready for service \( s \) until it can start service \( s' \).
- \( \hat{g}_w \): Number of potential passengers for demand \( w \).
- \( g_{\omega,s}^o \): Number of passengers for demand \( \omega \) who use the service \( s \).
- \( b_{\omega,s} \): Price of the ticket for demand \( \omega \) using service \( s \).
- \( K_s \): Capacity of service \( s \).
- \( \hat{K}_s \): Effective capacity of service \( s \).
- \( \hat{\pi}_s \): Effective operating costs of service \( s \).

Table 6.2: Complete notation used

the complete timetable assuming that initial times, speeds and stop times at stations are known. Some assumptions are made about the generic HSR system:

**Assumption 1.** Double-tracked line. Each section is separated by nodes (stations) and all the trains in the same section must travel in the same direction.
Assumption 2. The route of each service is fixed and defined by a sequence of stations. It isn’t necessary to stop at all stations. A function \( j = J(s, m) \) which indicates the \( m \)-th station visited by service \( s \) is assumed to be known. Also a dummy station \( j_F \), which represents the end of the service, is defined.

Assumption 3. The speed of each train is fixed while there is no type of conflict with another train. In case of conflict, the preceding train must reduce its speed maintaining the minimum dwell time defined by controllers in each track segment.

Assumption 4. A train which does not stop in a station could overtake a train stopped in this station. This constraint allows the system to operate express services which do not stop in each station they go through, contributing to the improvement in the performance of the system.

Two of the most important factors in the competitiveness of HSR systems are reduced travel times and punctuality. Because of this, taking into account the departure time and speed it is possible to consider that the only time added to the travel time of the trains is the necessary stopped time in stations, so the following assumptions are made:

Assumption 5. A train can stop only the time necessary for passenger alighting/boarding, without considering the possibility of connections between trains.

Assumption 6. There is no priority between trains at the stations, following a First Come First Served (FCFS) policy.

Thus the essential variables of the problem are the departure times of each train from their initial station. The supply model aims to determine the complete timetable of the HSR network.

The supply model receives the train departure times and simulates their behaviour taking into account the operational constraints. Finally the complete free-conflict timetable is calculated.

6.2.1.1 Dwell time and overtaking constraints

We denote by \( T_s \) the start time of service \( s \in S \) and we call

\[
T = (T_1, \cdots, T_n)
\]

the vector which contains the initial departure times of planned services, where \( n \) is the cardinal of the set \( S \).
6.2. A model for HSR scheduling

Each train $k \in K$ is constrained to a feasible number of services in a determined period. To describe the model, denote as $s = S(k,m)$ a function capable of obtaining the $m$th service, that is $s$, of train $k$.

**Definición 6.2.1.** An event $e$ represents the instant in which a train $k$ performing service $s$ leaves a station $j$ and is defined by the triplet:

$$e = (k, s_k, j_k)$$

where $k$ is a train, $s_k$ and $j_k$ are respectively the ordinal number of the service and the position of the station in which train $k$ stops. The respective service and station are obtained as follows:

$$s = S(k, s_k) \quad (6.1)$$

$$j = J(s, j_k) \quad (6.2)$$

The simulation model updates the events produced in the system dynamically. Figure 6.2 shows a real railway network at a certain instant. The example represents three trains with their associated events to be processed.

![Figure 6.2: Queue system represented as events](image)

The event instants $\delta_e$ correspond with the departure from a station, and determine the processing sequence of all the events.

Therefore the basic implementation of the discrete event simulation model is shown in the flow chart depicted in Figure 6.3. To process each event it is necessary to consider the two main constraints of the system, the dwell time and the rolling stock. The dwell time constraint is related to a section of the track and models the movement of a train in a track segment. Consider Figure 6.4. It shows track segment $i \rightarrow j$ and two trains. Denote $T_{si}^d$ as the departure time of service $s$ performed by train $k$ from station $i$. Denote $T_{sj}^a$ as the instant in which service $s$ arrives at station $j$. The system must satisfy:

$$T_{sj}^a \geq \tau_j^a + d_{ij} \quad (6.3)$$

$$T_{sj}^a \geq T_{si}^d + c_{ij} \quad (6.4)$$

where $d_{ij}$ is the dwell time of section $i \rightarrow j$, $\tau_j^a$ is the instant when the last train with respect to the previous moment of the system arrives at station $j$.
Equation (6.3) imposes a constraint which could lead a speed reduction for maintaining the minimum security time between trains. Equation (6.4) indicates the minimum instant $T_{a,sj}$ in which train $k$ may arrive at station $j$ considering the departure time $T_{d,si}$ from station $i$.

Taking into account both constraints (6.3) and (6.4) and that the TTP is focused on an HSR system, the main objective is to reduce the travel time to the minimum. Therefore:

$$T_{a,sj} = \max\{T_{d,si} + c_{ij}^k, \tau_{sj}^a + d_{ij}\}$$  \hspace{1cm} (6.5)$$

Moreover, it is necessary to consider the stop times of services:

$$T_{d,sj} \geq T_{a,sj} + p_{sj}$$  \hspace{1cm} (6.6)$$

where $p_{sj}$ is the stop time of service $s$ in station $j$. This approach represents the behaviour of the trains, following a FCFS policy when a train makes a stop in a station.

The model considers an overtaking situation in stations. If a train does not stop in a given station, it could pass the stopped trains in this station. Next, this situation is analysed for service $s$. 

---

**Figure 6.3:** Discrete event simulation model flow chart
6.2. A model for HSR scheduling

Denote by \( \tau_j^d \) the programmed departure instant from station \( j \) before the arrival of service \( s \) at station \( j \). The overtaking situation is addressed by calculating:

\[
\tau_j^d = \min \{ \tau_j^d, T_{s_j}^d \}
\]  

(6.7)

Finally when train \( k \) departs from station \( j \), the value of \( \tau_j^d \) is updated with the value \( T_{s_j}^a \).

### 6.2.1.2 Rolling stock constraint

This constraint considers that a planned service for a train cannot start until the previous service of this same train has ended, with a minimum security time and set up time between them. Suppose there is a train which carries out services \( s \) and \( s' \) consecutively, then:

\[
\tilde{T}_{s'} = \max \{ T_{s'}, \delta_s + r_{ss'} \}
\]

(6.8)

where \( \tilde{T}_{s'} \) is the real instant when \( s' \) starts, \( T_{s'} \) is the planned initial time, \( \delta_s \) is the end time of service \( s \) and \( r_{ss'} \) is the rolling stock turnaround time until the train can start service \( s' \).

### 6.2.1.3 The discrete event algorithm

The discrete event algorithm selects the next event to be processed by determining the train \( k \) associated with this event and calculates the arrival time at the next station \( j \) given the operational constraints. The arrival of
train $k$ at the station $j$ updates the last arrival time $\tau^a_j$ at this station, allowing overtaking if the constraints associated with $\tau^d_j$ are met.

The algorithm ends by generating the new event $e^*$ of train $k$, considering the rolling stock constraints and ending the service.

The complete discrete event algorithm can be seen in Table 6.3.

1. (Parameter initialization). $E = \{\emptyset\}$ and $\tau^a_j = -\infty$, $\tau^d_j = -\infty$ for each $j \in J$.
2. (Event initialization). For each $k \in K$, $s_k = 1$ and $j_k = 1$. Add to set $E$ the events $e = (k, 1, 1)$ for all $k$. Consider $T^d_{sj} = T_s$ (6.9) $\delta_e = T_s$ (6.10)
3. (Simulation). While $E \neq \{\emptyset\}$
   (a) Calculate next event. Consider $\delta^* = \min\{\delta_e\}$ (6.11) $e^* = \arg\min\{\delta_e\}$ (6.12)
   (b) Consider $e^* = (k, s_k, j_k)$, compute next event for train $k$. Define $i = J(k, j_k)$ and $j_k = j_k + 1$. Calculate $j = J(k, j_k)$, there are three possibilities:

   i. (The service has more stops). If $j_k \neq j_F$ then:
      $T^a_{sj} = \max\{T^a_{dsi} + c^k_{ij}, \tau^a_j + d_{ij}\}$ (6.13)
      $T^d_{sj} \geq T^a_{sj} + p_{sj}$ (6.14)
      $\tau^d_j = \min\{\tau^d_j, T^a_{sj}\}$ (6.15)
      $\tau^d_j = T^a_{sj}$ (6.16)

      Compute new event $e^* = (k, s_k, j_k)$ and $\delta^*_e = T^d_{sj}$.

   ii. (Actual service has ended, but there are more services for the train). If $j_k = j_F$ and $S(k, s_k + 1) \neq s_F$, then: $j_k = 1, s = S(k, s_k + 1), s_k = s_k + 1, s^' = S(k, s_k + 1)$, and $j = J(s, 1)$. Taking into account the Rolling Stock constraint:
      $T^d_{s^', j} = \max\{T^d_{s^', s^s}, \delta^*_e + r_{ss^'}\}$ (6.17)

      Compute new event $e^* = (k, s_k, 1) \mathcal{y} \delta^*_e = T^d_{s^', j}$.

   iii. (There are no more events for train $k$). Delete event $e^*$, $E = E - \{e^*\}$.

---

**Table 6.3**: Discrete event simulation algorithm
6.2. A model for HSR scheduling

6.2.2 Demand model and utilities

Chapter 5 describes the HSR demand model used and the process by which it was calibrated.

6.2.3 Objective function

This model considers the maximization of profit, that is, income of ticket sales minus operating costs.

The income of ticket sales is defined as $\sum_{\omega \in W} \sum_{s \in S_{\omega}} g_{\omega,s}^{at} b_{\omega,s}$, where $g_{\omega,s}^{at}$ is the number of passengers of demand $\omega$ who use the service $s$ and $b_{\omega,s}$ is the price of the ticket for a demand $\omega$ using service $s$.

Let $\pi_s$ be the operating costs of service $s$ and $K_s$ the capacity of the train that operates service $s$, we define the effective capacity as:

$$\tilde{K}_s(T_s) := \begin{cases} K_s & \text{if } T_s < T \\ 0 & \text{otherwise} \end{cases} \quad (6.18)$$

and the effective operating cost as:

$$\tilde{\pi}_s(T_s) := \begin{cases} \pi_s & \text{if } T_s < T \\ 0 & \text{otherwise} \end{cases} \quad (6.19)$$

These definitions represent the fact that if a service $s$ is operated in the planned period $[0, T]$ then the cost of this service is $\pi_s$ with a capacity $K_s$. Otherwise, the operating cost of this service is 0 and it cannot serve the demand.

Thus the objective function is stated as:

$$\text{Maximize } T \cdot Z(T) = \sum_{\omega \in W} \sum_{s \in S_{\omega}} g_{\omega,s}^{at} \left( \tilde{K}_s(T_s) \right) b_{\omega,s} - \sum_{s \in S} \tilde{\pi}_s(T_s) \quad (6.20)$$

It is worth noting that if a service $s$ satisfies $T_s > T$ then it is not profitable to operate this service. If $T = +\infty$ all the planned services will be scheduled, and the operating costs of the HSR network will be constant with respect to the decision vector $T$. In this case the intention is to align the timetable with the desired trip of each user.

Furthermore, note that the fact that resolution algorithms are based on objective function evaluations means that definitions (6.18) and (6.19) do not present problems.
<table>
<thead>
<tr>
<th>Type</th>
<th>Average Speed (km/h)</th>
<th>Capacity ($K$)</th>
<th>Operative Cost (euros)</th>
<th>Cost/km</th>
</tr>
</thead>
<tbody>
<tr>
<td>AVE</td>
<td>215</td>
<td>308</td>
<td>17.5</td>
<td></td>
</tr>
<tr>
<td>AVANT</td>
<td>193</td>
<td>237</td>
<td>16.4</td>
<td></td>
</tr>
<tr>
<td>MD</td>
<td>106</td>
<td>190</td>
<td>10.2</td>
<td></td>
</tr>
</tbody>
</table>

Table 6.4: Features of each service on Madrid-Seville corridor

### 6.3 Metaheuristic algorithm

The optimization model shown in Figure 6.1 represents a bi-level structure. The evaluation of the objective function needs: i) to run a simulation model to calculate the complete timetable (Table 6.3), ii) to evaluate the non-linear utilities and iii) to solve a convex program for estimating the demand disaggregation per service. Therefore it is not possible to use exact methods and free derivative methods must be used. In this case the hybrid approach described in Chapter 4 is followed using the SPSO+NM algorithm of Chapter 5. Furthermore the evaluation of the objective function of the proposed HSR-TTP model has a significant computational cost. This has motivated the use of parallelization strategies to take advantage of the multiple processors of modern computers to reduce the computational burden of the algorithm. This is a new feature with respect to the implementation used in Chapter 5.

### 6.4 Computational experiments

This section tries to test the proposed methodology based on free derivative methods focusing on two main objectives: i) studying the quality of the solution found and ii) study the computational efficiency of the algorithms.

This model has been implemented in a software application called dhAVE and run in MATLAB 2013, while the demand model has been solved using CONOPT. The computer used to perform the computational experiments has the following characteristics: Windows 7 64 bits, processor: 2 x AMD Opteron 4226, 12 cores 2.7GHz, RAM: 12GiB 1600MHz.

#### 6.4.1 Case study

To test the HSR-TTP model, a the complete case study focused on Madrid-Seville corridor and presented in Chapter 5 is considered. Table 6.4 complements the data used showing the characteristics of each type of train.

The origin-destination matrix (different potential trips between stations) on a working day is shown in Table 6.5.
6.4. Computational experiments

<table>
<thead>
<tr>
<th></th>
<th>MA</th>
<th>CR</th>
<th>PU</th>
<th>COR</th>
<th>SEV</th>
</tr>
</thead>
<tbody>
<tr>
<td>MA</td>
<td>-</td>
<td>1377</td>
<td>479</td>
<td>185</td>
<td>3270</td>
</tr>
<tr>
<td>CR</td>
<td>1377</td>
<td>-</td>
<td>216</td>
<td>185</td>
<td>185</td>
</tr>
<tr>
<td>PU</td>
<td>479</td>
<td>216</td>
<td>-</td>
<td>185</td>
<td>185</td>
</tr>
<tr>
<td>COR</td>
<td>185</td>
<td>185</td>
<td>185</td>
<td>-</td>
<td>1291</td>
</tr>
<tr>
<td>SEV</td>
<td>3270</td>
<td>185</td>
<td>185</td>
<td>1291</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 6.5: Potential total trips ($\hat{g}_w$)

The baseline in order to compare the timetable obtained consists of the current schedule used by RENFE after running the current planned schedule in the simulation module. The simulated time-station diagram is shown in Figure 6.5. To compare the solution obtained with the actual schedule of RENFE avoiding distortions, the operation of all services has been forced, this is $T = +\infty$.

![Time-station graph obtained by simulation](image)

Figure 6.5: Time-station graph obtained by simulation

The demand model computes the utility for each train and each demand. In this example only one type of user ($\ell = 1$) is considered with regard to economic characteristics, but taking into account the trip of each traveller (origin-destination pair $\omega$), there will be 20 types of user. Each user type $\omega$ can travel using a set of services $s \in S_\omega$. Considering the planned schedule, the set of alternatives $(\omega, s) \in A$ consists of 298 alternatives.

The baseline value is calculated using the current RENFE schedule with an objective value of 364180 euros per day.
6.4.2 Experiment I: qualitative analysis of solution

This section shows a set of 11 experiments using the NM, SPSO and SPSO+NM algorithms. Each algorithm has been run for 1000 objective function evaluations. NM algorithm starts from the planned schedule to obtain its proposed solution. SPSO and SPSO+NM methods start with an initial swarm of 40 particles calculated randomly on an interval [0:00, 23:59] hours. The parameters $w$, $c_1$ and $c_2$ for updating velocity are defined as $w = 1/(2 \ln(2))$, $c_1 = 0.5 + \ln(2)$ and $c_2 = c_1$ (see Zambrano-Bigiarini et al. (2013)). SPSO+NM values $n_c$ and $n_r$ are set to 1, 5, 25 and 1, 5, 10% of the total iterations respectively, combining all the possibilities.

The results obtained are shown in Table 6.6. The first column indicates the algorithm used and its configuration. Also the solution found and the CPU time of the algorithm can be shown in the second and third columns respectively. The last column contains the improvement obtained with respect to the baseline value using the following equation:

$$\text{Improvement} = \frac{Z - \text{baseline value}}{\text{baseline value}} \times 100$$

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Solution found (€)</th>
<th>CPU time (h)</th>
<th>Improvement (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>NM</td>
<td>367836</td>
<td>6.3</td>
<td>1.00</td>
</tr>
<tr>
<td>SPSO</td>
<td>371642</td>
<td>6.1</td>
<td>2.04</td>
</tr>
<tr>
<td>SPSO+NM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>($n_c = 1$, $n_r = 1$)</td>
<td>373130</td>
<td>6.1</td>
<td>2.45</td>
</tr>
<tr>
<td>($n_c = 1$, $n_r = 5$)</td>
<td>372833</td>
<td>6.6</td>
<td>2.37</td>
</tr>
<tr>
<td>($n_c = 1$, $n_r = 10$)</td>
<td>371904</td>
<td>7.5</td>
<td>2.12</td>
</tr>
<tr>
<td>($n_c = 5$, $n_r = 1$)</td>
<td>372032</td>
<td>6.1</td>
<td>2.15</td>
</tr>
<tr>
<td>($n_c = 5$, $n_r = 5$)</td>
<td>372344</td>
<td>6.6</td>
<td>2.24</td>
</tr>
<tr>
<td>($n_c = 5$, $n_r = 10$)</td>
<td>372816</td>
<td>7.0</td>
<td>2.37</td>
</tr>
<tr>
<td>($n_c = 25$, $n_r = 1$)</td>
<td>372709</td>
<td>6.1</td>
<td>2.34</td>
</tr>
<tr>
<td>($n_c = 25$, $n_r = 5$)</td>
<td>372228</td>
<td>6.6</td>
<td>2.20</td>
</tr>
<tr>
<td>($n_c = 25$, $n_r = 10$)</td>
<td>373160</td>
<td>7.2</td>
<td>2.46</td>
</tr>
</tbody>
</table>

Table 6.6: Improvements obtained by NM, SPSO and SPSO+NM algorithms

As can be seen, the computational time of the optimization algorithms is between 6.1 and 7.5 hours producing an improvement percentage around 1 in the NM case and more than 2 in SPSO-related algorithms. It is worth noting that the SPSO+NM approach obtained in all cases a better solution than NM and SPSO algorithms alone.

Moreover, Figure 6.6 shows the objective value versus the number of evaluations of the objective function.
6.4. Computational experiments

6.4.3 Experiment II: test of parallelization strategies

This section analyses the possibility of reducing the computational time of the proposed optimization algorithms. SPSO and SPSO+NM algorithms can be improved because parts of them can be parallelized. There is software code that can be changed from sequential (see Figure 6.7) to parallel mode (see Figure 6.8). This part then changes its behaviour and several evaluations of the objective function can be done at the same time.

Using this idea, the speed up-ratio is obtained for each algorithm using the following equation:

$$ speed \uparrow \text{ratio}_n = \frac{\text{Sequential runtime}}{\text{Parallel runtime with } n \text{ cores}} $$

(6.22)

Furthermore, the parallelization of MATLAB-GAMS connected programs is not trivial. This matter have been addressed in Appendix C.

This experiment has been carried out running the SPSO and SPSO+NM
Figure 6.7: Sequential SPSO

\((n_c = 25, n_r = 10)\) algorithms in parallel mode with 2, 4, 8 and 12 parallel threads. This configuration of the algorithm has been selected because it is the one which found the better solution. The results of the computational cost and speed-up ratio of the algorithms can be seen in Table 6.7 and Figure 6.9.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>CPU time (hours)</th>
<th>Speed up-ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPSO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2 cores</td>
<td>3.5</td>
<td>1.74</td>
</tr>
<tr>
<td>4 cores</td>
<td>1.8</td>
<td>3.39</td>
</tr>
<tr>
<td>8 cores</td>
<td>1.1</td>
<td>5.54</td>
</tr>
<tr>
<td>12 cores</td>
<td>1.0</td>
<td>6.10</td>
</tr>
<tr>
<td>SPSO+NM (n_c = 25, n_r = 10)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2 cores</td>
<td>3.7</td>
<td>1.94</td>
</tr>
<tr>
<td>4 cores</td>
<td>1.8</td>
<td>4.00</td>
</tr>
<tr>
<td>8 cores</td>
<td>2.1</td>
<td>3.43</td>
</tr>
<tr>
<td>12 cores</td>
<td>2.0</td>
<td>3.60</td>
</tr>
</tbody>
</table>

Table 6.7: Computational comparison between parallel algorithms

The results show the that improvement of each algorithm depends on
6.4. Computational experiments

Figure 6.8: Parallel SPSO

Figure 6.9: Speed-up ratio for SPSO and SPSO+NM algorithms

the number of parallel threads. However is worth noting that the speed up-ratio in the SPSO+NM algorithm does not increase with the number
of parallel threads. This occurs because of the random nature of the algorithm, in which the NM algorithm, which is non-parallelizable, was ran an indeterminate number of times depending on the solutions found by the SPSO stage. The case with 2 and 4 parallel threads only ran the NM algorithm once, performing 1000 function evaluations, while with 8 and 12 cores the NM algorithm is run twice, performing 2000 function evaluations.

In conclusion, the best algorithm is the SPSO+NM hybridization which achieves an improvement of 2.46% and the parallelization strategies are capable of reducing the computational time of this algorithm to 2 hours.

6.5 Conclusions

This Chapter proposes a disaggregated multi-modal schedule-based HSR model focused on endogenous factors, to determine the HSR demand. This model is divided into two modules. The first is a discrete event simulation model which calculates the whole timetable considering the departure time of the trains from the origin station and several operational constraints. After this, a novel constrained nested logit model obtains the number of passengers that will use the services based on the calculated departure times, the travel times, the ticket price and the capacity constraints of each service.

The proposed HRS-TTP model aims at maximizing the profits of the railway operator. It is solved using derivative free algorithms, hybridization techniques and parallelization techniques in a computational experiment based on real data from the Madrid-Seville high-speed corridor.

The results shows the applicability of the proposed approach to a real scenario in an acceptable computational time and how the model generated represents the behaviour of the users of the railway network considering the timetable calculated.
7. Conclusions and future research
Conclusions and future research

7.1 Conclusions

The conclusions obtained during the development of this research have been stated at the end of each chapter. In this section the main conclusions reached in this thesis are summarized.

This PhD thesis studies and proposes two methodologies for solving the railway scheduling problem. The first deals with railway rescheduling in an on-line context, the other one addresses the off-line context.

Considering the on-line context, the train rescheduling problem is addressed by proposing an ITS composed of a conflict detection module and a conflict resolution module. The first tool is defined as a discrete event simulation model which represents the behaviour of the trains in the railway system, and the second is implemented using alternative graphs and solved with algorithms from the literature (FCFS and AMCC), and a new approach focused on the reduction of total delays which might consider the demand if it is known.

Several numerical experiments have been carried out on RENFE Cercanías Madrid network. These results show that this methodology can be applied in a real-time context and can improve the schedule, detecting future conflicts and solving them optimally.

Focusing on the off-line context, the TTP is addressed proposing a HSR-TTP model that integrates supply and demand equilibrium. The supply is represented by a discrete event mesoscopic simulation model which considers the operational constraints of the railway network, and the demand behaviour using a novel constrained nested logit model (CMNL) based on a random utilities framework. Moreover, a new approach has been developed for the hybridization of derivative free algorithms, which are used
to calibrate the CMNL and solve the proposed HSR-TTP model.

The results of the study carried out on the Madrid-Seville high-speed corridor show the models are able to simulate and represent the real railway system’s characteristics accurately while the computational time using the techniques proposed is affordable in our planning context.

Furthermore it is worth noting that the two methodologies proposed in this thesis might change the mode of transport used between regional or HSR because they are not strongly tied to the problem represented.

Also is necessary to emphasize the fact that the structure of this research, dividing the approaches into various modules, allows new models and algorithms to be developed and implemented that can replace any part of those proposed with the objective of testing new approaches which could improve the existing ones.

### 7.2 Suggestions for further research

This PhD thesis is the result of a learning process in research into optimization models and algorithms to approach the railway scheduling problem. This work opens several lines for further research.

In this thesis new methods for railway scheduling based on optimization models, discrete event simulation, metaheuristic techniques, parallel computation, discrete choice models, Hilbert space theory, etc. have been developed. All these concepts can be put together with the objective of creating new methods which can deal with other applications. These new applications present a new challenge, generating new methods, closing a feedback mechanism, which is the engine of an endless machine called research. This is the point of view of this thesis which it is intended to continue. For this reason, the future lines of research are divided into i) applications and ii) development of new mathematical methods.

- **Applications**

  * To establish an ITS standard.

  In this PhD thesis the basis of a railway ITS are defined. This is an aspect that will be tackled by implementing a complete ITS prototype.

  The models proposed in this thesis can be extended by considering more control strategies. Specifically, this thesis has dealt with the scheduling problem but other control strategies might be included, such as speed change, re-routing or service cancellation. Depending on the problem to be solved, it is important to define correctly the order of action or priority for
7.2. Suggestions for further research

Each strategy. Furthermore, it is possible to extend the models by changing the predefined modules with previous or future approaches defined in the literature.

Therefore, following all these ideas, a standard for developing ITS might be defined to implement future ITS prototypes or software tools.

* Case study extensions.

One of the facts that must be considered when defining a model for railway transportation is to take into account the differences between the railway systems implemented around the world. In this case, to evaluate the viability of using the proposed models in other railways it would be interesting to test these models and algorithms with the networks from other countries. The main problem is to obtain the necessary data.

* Consider the preferences of users in microscopic network modelling.

Another interesting extension is to test the introduction of a demand model in the TTP approach for analysing a regional railway network defined as a microscopic model like the one developed for the ITS. A first study in this area will be presented in Espinosa-Aranda et al. (2014a) in which the users’ choice of departure time is considered and an integer linear programming model is given.

* Timetable design under uncertainty.

The TTP models are applied in an operative planning level, and the uncertainty can be considered negligible. The models described in this thesis might also be used for long-term planning such as combined models for rolling stock acquisition, and the accuracy in this case is open to question. Thus, the robustness of the models could be improved by considering non-deterministic scenarios.

* Development of new mathematical methods

* New hybridizations in global optimization.

The metaheuristic algorithms developed in Chapter 4 constitute a widely algorithmic class and only a few instances of the generic algorithms have been used, and a more exhaustive study must be performed. Several of the global optimization methods developed try to find an equilibrium between the exploration of the space of solutions and the exploitation of the most promising zones. The numerical comparison carried out in Rios and Sahinidis (2012) shows that one of the current better methods is to use TOMLAB/GLCLUSTER. This study suggests that cluster analysis might be a powerful tool for determining which solution set belongs to the same neighbourhood. This information might allow the different exploration and exploitation phases to be managed correctly. In this thesis the parameter $n_c$
has been considered only as a tool for changing between the exploration and exploitation phases.

Therefore, other parallelization methods might be studied, like the use of a computer’s multiprocessor graphic cards and CUDA technologies.

* Simulation-based optimization

Simulation-based optimization is an emerging field which integrates optimization techniques into simulation analysis. The HSR-TTP model presented in this thesis is an example of this approach. The characteristics of the objective function are computationally expensive to evaluate and not necessarily differentiable.

In the literature, the *Response Surface Methodology* (RSM) has been applied to simulation-based optimization (see Jakobsson et al. (2010b), Jakobsson et al. (2010a)). The so-called *Mercer Kernel* allows us to obtain the *surrogate model* from a set of points where the objective function has been evaluated. The optimization of a *quality function*, based on the surrogate model and the current set of points, generates a new candidate which improves the current surrogate model. In this thesis *Mercer Kernels* have been used in the modelling of dynamic utilities, but they can also be considered as a powerful tool to be used in RSM to optimize the HSR-TTP model or for the calibration of the CMNL model.
Bibliography


Bibliography


In this Appendix are listed the used functions in the experiments carried out in Chapter 4. These tests are obtained from Fan and Zahara (2007). We have adopted the following format:

Function Number) Function name

(a) Function Definition
(b) Global optimum

2D Functions

1) Powell badly scaled function

(a) \( f(x) = (10x_1x_2 - 1)^2 + (\exp[-x_1] + \exp[-x_2] - 1.0001)^2 \)
(b) \( f^* = 0 \) at \((1.098...e-5,9.106...)\)

2) B2 Function

(a) \( f(x) = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1) - 0.4\cos(4\pi x_2) + 0.7 \)
(b) \( f^* = 0 \) at \((0,0)\)

3) Beale Function
(a) $f(x) = \sum_{i=1}^{3} (y_1 - x_1 (1 - x_i^2))^2$ where $y_1 = 1.5$, $y_2 = 2.25$, $y_3 = 2.625$

(b) $f^* = 0$ at $(3.0.5)$

4) Booth Function

(a) $f(x) = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2$

(b) $f^* = 0$ at $(1.3)$

3D Functions

5) Helical valley function

(a) $f(x) = (10[x_3 - 10\theta(x_1, x_2)])^2 + (10[(x_1^2 + x_2^2)^{1/2} - 1])^2 + x_3^2$ where

$$\theta(x_1, x_2) = \begin{cases} 
\frac{1}{\pi}\text{arctan}\left(\frac{x_1}{x_2}\right) & \text{if } x_1 > 0 \\
\frac{1}{2\pi}\text{arctan}\left(\frac{x_1}{x_2}\right) + 0.5 & \text{if } x_1 < 0 
\end{cases}$$

(b) $f^* = 0$ at $(1.0.0)$

6) De Jong Function

(a) $f(x) = x_1^2 + x_2^2 + x_3^2$

(b) $f^* = 0$ at $(0.0.0)$

7) Box three-dimensional function

(a) $f(x) = \sum_{i=1}^{3} (\exp[-t_i x_1] - \exp[-t_i x_2] - x_3 (\exp[-t_i] - \exp[-10t_i]))^2$ where $t_i = (0.1)i$

(b) $f^* = 0$ at $(1.10.1)$, $(10,1,-1)$ and wherever $x_1 = x_2, x_3 = 0$
4D Functions

8) Wood Function

\[ f(x) = (10(x_2 - x_1^2) + (1 - x_1)^2 + (90^{1/2}(x_4 - x_3^2))^2 + (1 - x_3)^2 + (10^{1/2}(x_2 + x_4 - 2))^2 + (10^{-1/2}(x_2 - x_4))^2 \]

(b) \( f^* = 0 \) at \((1,1,1,1)\)

9) Trigonometric Function

(a) \( f(x) = \sum_{i=1}^{4} f_i^2(x) \) where \( f_i(x) = 4 - \sum_{j=1}^{4} \cos x_j + i(1 - \cos x_i) - \sin x_i, \quad i = 1, 2, 3, 4 \)

(b) \( f^* = 0 \)

10) Rosenbrock Function

(a) \( f(x) = \sum_{i=1}^{2} (100(x_2i - x_2^{2i-1})^2 + (1 - x_2i-1)^2) \)

(b) \( f^* = 0 \) at \((1,1,1,1)\)

11) Variably dimensioned function

(a) \( f(x) = \sum_{i=1}^{4} (x_i - 1)^2 + (\sum_{i=1}^{4} i(x_i - 1))^2 + (\sum_{i=1}^{4} i^2(x_i - 1))^4 \)

(b) \( f^* = 0 \) at \((1,1,1,1)\)

8D Functions

12) Penalty function I

(a) \( f(x) = \sum_{i=1}^{8} ((10^{-5})^{1/2}(x_i - 1))^2 + ((\sum_{j=1}^{8} x_j^2) - \frac{1}{4})^2 \)

(b) \( f^* = 5.42152e - 05 \)

13) Penalty function II

(a) \( f(x) = \sum_{i=1}^{16} f_i^2(x) \) where
\[ f_1(x) = x_1 - 0.2 \]
\[ f_i(x) = a^{1/2} \exp[x_i/10] + \exp[x_{i-1}/10] - y_i, \quad 2 \leq i \leq 8 \]
\[ f_i(x) = a^{1/2} \exp[x_{i-n+1}/10] - \exp[(-1)/10]), \quad 8 < i < 16 \]
\[ f_{16}(x) = (\sum_{j=1}^{8} (8 - j + 1)x_j^2) - 1 \]
\[ a = 10^{-5}, \quad y_i = \exp[i/10] + \exp[(i - 1)/10] \]
\[ (b) \quad f^* = 1.23335\times10^{-04} \]

14) Trigonometric Function (II)

(a) \[ f(x) = \sum_{i=1}^{8} f_i^2(x) \] where \[ f_i(x) = 8 - \sum_{j=1}^{8} \cos x_j + i(1 - \cos x_i) - \sin x_i \]
\[ i = 1, 2, \ldots, 8 \]
\[ (b) \quad f^* = 0 \]

15) Extended Powell Function

(a) \[ f(x) = \sum_{i=1}^{8} f_i^2(x) \] where
\[ f_{4i-3}(x) = x_{4i-3} + 10x_{4i-2}, \quad i = 1, 2 \]
\[ f_{4i-2}(x) = 5^{1/2}(x_{4i-1} - x_{4i}), \quad i = 1, 2 \]
\[ f_{4i-1}(x) = (x_{4i-2} - 2x_{4i-1})^2, \quad i = 1, 2 \]
\[ f_{4i}(x)10^{1/2}(x_{4i-3} - x_{4i})^2, \quad i = 1, 2 \]
\[ (b) \quad f^* = 0 \] at origin

10D Functions

16) Griewank Function

(a) \[ f(x) = \sum_{i=1}^{10} \frac{x_i^2}{4000} - \prod_{i=1}^{10} \cos \frac{x_i}{\sqrt{i}} + 1 \]
\[ (b) \quad f^* = 0 \] at origin

17) Rastrigin Function
(a) \( f(x) = \sum_{i=1}^{10} (x_i^2 - 10 \cos(2\pi x_i) + 10) \)

(b) \( f^* = 0 \) at origin

18) Rosenbrock Function

(a) \( f(x) = \sum_{i=1}^{5} (100(x_{2i} - x_{2i-1}^2)^2 + (1 - x_{2i-1})^2) \)

(b) \( f^* = 0 \) at \((1,1,1,1,1,1,1,1,1,1)\)

30D Function

19) Sphere Function

(a) \( f(x) = \sum_{i=1}^{30} x_i^2 \)

(b) \( f^* = 0 \) at origin

50D Function

20) Griewank Function (II)

(a) \( f(x) = \sum_{i=1}^{50} \frac{x_i^2}{4000} - \prod_{i=1}^{50} \cos \frac{x_i}{\sqrt{i}} + 1 \)

(b) \( f^* = 0 \) at origin
Case studies data and links

In this Appendix, several links for case studies and videos are given.

B.1 Chapter 2

Videos of SIMEIFER tool:


B.2 Chapter 3

Data concerning the Madrid regional network and the complete disturbance scenario:


Video showing the working of SOFGA tool:

http://youtu.be/771\_wAqcphI

File with the generated data of demand variable $g_i$:

B.3 Chapters 5 and 6

Madrid-Seville corridor simulation and calibrated data for optimization:

   http://bit.ly/1gCFw5e

Videos of dhAVE tool:

   dhAVE tool normal usage: http://youtu.be/NmUDtzOFmCc
   GAMS parallelization: http://youtu.be/DigCgL9h9xA
MATLAB-GAMS parallelization

MATLAB is a software which brings lots of toolboxes and functions for mathematical computation which includes an optimization toolbox. This library has limited options for the formulation mathematical modelling. In contrast, modelling languages such as GAMS have this capability, and have been used in many practical applications, but lacks some capabilities for data visualization and manipulation, where MATLAB is better. This fact justifies the objective of linking MATLAB and GAMS to combine the best features of each language.

In Ferris (1998) GDXMRW library is implemented. This library gives a suite of utilities to import/export data between GAMS and MATLAB and to call GAMS models from MATLAB getting the results back in MATLAB. The software is intended to give MATLAB users the ability to use all the optimization capabilities of GAMS, and allows visualization of GAMS models directly within MATLAB in an easy manner so that optimization results can be viewed and processed using any of the toolboxes that exist in MATLAB.

The installation procedure of GDXMRW can be seen at Ferris et al. (2011). The most recent version (GAMS Distribution 23.4) includes GDXMRW library.

Simulation-based models, like the one proposed in Chapter 6 of this PhD thesis, require to compute a complete simulation which can not be implemented in GAMS, so an imperative programming language like MATLAB must be used. After that, an optimization model which represents another part of the whole model (in this case the CMNL model) can be solved with an exact algorithm in a acceptable computational time, thus GAMS is used.

These simulation-based models can be solved using free derivative methods. Within this type of algorithms several of them are based on the evaluation of the objective function, and furthermore, it is possible to reduce their computational time using parallelization techniques in high
performance architectures using multiple threads.

The GDXMRW library is able to be executed in a parallel mode and solve this problem, but the process is not explained in the literature or library manual's. This section explain how this situation has been tackled with and implement a tool capable of executing various instances of a GAMS model with different data using the same model at the same time automatically from MATLAB.

C.1 Problems

The basic instruction from MATLAB to call a model implemented in GAMS is as follows:

```matlab
[x1, x2, ..., xn] = gams('model', s1, s2... sm);
```

where \(x_1 \cdots x_n\) are the output structures containing the output sets or variables, \(model\) is the name of the .gms file to be executed and \(s_1 \cdots s_m\) are the input structures containing the output sets or variables. Next, a simple example of a GAMS code adapted to be executed from MATLAB and taken from Ferris et al. (2011) is shown:

```gams
$set matout " matsol.gdx", x, dual ";
set i /1*2/;
set j /1*3/;
alias (j1, j);
parameter
Q(j,j1) /
  1 .1 1.0
  2 .2 1.0
  3 .3 1.0 /,
A(i,j) /
  1 .1 1.0
  1 .2 1.0
  1 .3 1.0
  2 .1 -1.0
  2 .3 1.0 /,
b(i) /
  1 1.0
  2 1.0 /,
c(j) /
  1 2.0 /
$if exist matdata.gms $include matdata.gms
variable obj;
positive variable x(j);
equation cost, dual(i);
cost.. obj =e= 0.5*sum(j, x(j)*sum(j1, Q(j,j1)*x(j1))) + sum(j, c(j)*x(j));
dual(i).. sum(j, A(i,j)*x(j)) =g= b(i);
model qp /cost, dual/;
solve qp using nlp minimizing obj;
execute_unload %matout%;
```
C.2. Solution proposed

The main code lines which interact with MATLAB are:

$set matout "'matsol.gdx', x, dual ";
...
$if exist matdata.gms $include matdata.gms
...
execute_unload %matout%;

First and last lines are focused on unload the output variables in matsol.gdx file (x and dual in this case) when the execution of the file ends, while second line is focused on loading the input variables at the start of the execution. As can be seen, if this example is executed several times in various concurrent threads, the main problem found will be the possibility of two or more threads writing/reading the .gdx files concurrently, consequently producing an error. Therefore, next section propose an approach for solving this problem.

C.2 Solution proposed

The prior problem can be solved generating several GMS files which do not write in matdata.gdx or read from matsol.gdx and combining the write and read tools of GDXMRW (wgdx and rgdx) with gams instruction without including any type of parameter directly.

C.2.1 GAMS code

The first step is to show how the GAMS code must be implemented. In this case, the instructions for loading the input variables at the start of the execution should appear before the VARIABLES definition as follows:

...$gdxin matdata1
$load var1, var2, ..., varn
$gdxin
VARIABLES
...

The text matdata1 refers to the file where the variables from MATLAB are written, while var1 to varn are the variables or sets that will be read from the file.

To unload the results of the model, the follow line must be written just at the end of the GAMS file:

...execute_unload 'matsol1.gdx', unl1, unl2, ..., unln;
*end of the file
Where *matsol1.gdx* is the file where GAMS will write the values of the variables or sets *unl1* to *unln* at the end of the GAMS execution, which will be read from MATLAB.

An example GAMS file called *logitCasoEstudio.gms* taken from the model implemented in Chapter 5 of this PhD thesis is as follows:

```plaintext
option solvelink=5;

SETS
J station
W demand
L type of users /1/
S services
M modes of transport /1, 2/
JS(S,J) stations per service
Wmas(W,S,J)
Wsube(W,S,J)
SW(W,S):

PARAMETERS
Gbar(W,L) demand
Ks(S) service capacity
beta1(L)
beta2(L)
etα(M,L)
Ua(W,S,L) utilities
etaC(L):

$gdxin matdata1
$load J
$load W
$load S
$load JS
$load Wmas
$load Wsube
$load SW
$load Gbar
$load Ks
$load beta1
$load beta2
$load Ua
$gdxin

eta('1',L)=1/beta1(L)−1/beta2(L):
etα('2',L)=1/beta1(L):
etαC(L)=1/beta2(L):

VARIABLES
Z objective function:

POSITIVE VARIABLES
G(W,S,M,L)
Gm(M,W,L):
```
C.2. Solution proposed

\[ \begin{align*}
G_{LO}(W,S,M,L) &= 0.1; \\
G_{mLO}(M,W,L) &= 0.1; \\
\text{EQUATIONS} \\
\text{FO} & \quad \text{objective function} \\
\text{Demanda1}(W,L) & \quad \text{demand1} \\
\text{Demanda2}(W,L) & \quad \text{demand2} \\
\text{Capacidad}(S,J) & \quad \text{service capacity}; \\
\text{FO} & \quad Z = \sum(W, \sum(L, \sum(M, \eta(M,L) \times G_{m}(M,W,L) \times \log(G_{m}(M,W,L) - 1)) \times C(L) \times \sum(S, SW(W,S) \times G(W,S,1',L) \times \log(G(W,S,1',L) - 1)) - \sum(S, SW(W,S) \times Ua(W,S,L) \times G(W,S,1',L)))) \\
\text{Demanda1}(W,L) & \quad \sum(M, G_{m}(M,W,L)) = \gamma(W,L); \\
\text{Demanda2}(W,L) & \quad \gamma(1',W,L) = \sum(S, SW(W,S) \times G(W,S,1',L)); \\
\text{Capacidad}(S,J) & \quad \sum(S, Ks(S,J)) = \sum(S, SW(W,S,1',J,L)) = 1 = Ks(S); \\
\text{MODEL nd} & /\text{ALL}/; \\
\text{SOLVE nd USING nlp MINIMIZING } Z; \\
\text{execute_unload } 'matsol1.gdx', Z, \text{Capacidad}, G, Gm, SW, Ua, beta1, beta2, W;
\end{align*} \]

\[ \begin{align*}
\text{C.2.2 MATLAB code} \\
\text{The code for connecting MATLAB with GAMS in parallel mode without} \\
\text{causing any type of error must include 3 main parts and one optional piece} \\
\text{of code.}
\end{align*} \]

First, the code should be capable of identifying the \textit{matdata1} and \textit{matsol1} texts within the GAMS code. The code for performing this idea with \textit{logitCasoEstudio.gms} file presented before can be seen afterwards:

\[ \begin{align*}
% \text{load original file} \\
\text{arch_orig} &= \text{textread(’logitCasoEstudio.gms’, '%s', 'delimiter', '
')); \\
\text{busq_matdata} &= \text{strfind(arch_orig, 'matdata1');} \\
\text{busq_matsol} &= \text{strfind(arch_orig, 'matsol1');} \\
\text{matdata} &= [0 0 0]; \\
\text{matsol} &= [0 0 0]; \\
% \text{save positions values for matdata1 and matsol1 texts} \\
\text{for } i = 1: \text{length(busq_matdata)} \\
\text{\quad if } \text{isempty(busq_matdata{1})} \\
\text{\quad \quad matdata} &= [i \text{ busq_matdata{1}}] \text{ busq_matdata{1}+7]; \\
\text{\quad end} \\
\text{\quad if } \text{isempty(busq_matsol{1})} \\
\text{\quad \quad matsol} &= [i \text{ busq_matsol{1}}] \text{ busq_matsol{1}+10]; \\
\text{\quad end} \\
\end{align*} \]

Next step is to generate as many files as the parallel code mights execute changing \textit{matdata1} and \textit{matsol1} by \textit{matdatan} and \textit{matsoln} respectively,
where \( n \) is the current number of the file created (in this case, logitCasoEstudio.gms):

```matlab
% generation of all the gms files (40 in this case)
particles=40;
for i=1:particles

    f=fopen(['logitCasoEstudio' num2str(1) '.gms'], 'wt');
    for j=1:length(arch_orig)
        if j==matdata(1)
            fprintf(f, ['$gdxin matdata ' num2str(i) '
']);
        elseif j==matsol(1)
            fila=arch_orig(j);
            fprintf(f, [fila(1:(matsol(2)-1)) 'matsol' num2str(i) '.gdx' fila
                    ((matsol(3)+1):length(fila)) '
']);
        else
            fprintf(f, [arch_orig(j) '
']);
        end
    end
    fclose(f);
end

Finally in each parallelized execution the input variables will be loaded into the respective GMS file using the \texttt{wgdx} function. Next, the appropriate GMS file is executed using \texttt{gams} function without any inputs or outputs and finally the results are retrieved using \texttt{rgdx} function. The following code is an example of this idea, which was included in the SPSO parallel algorithm presented in Chapter 6 and executed for 40 particles at each iteration:

```matlab
parfor i=1:particles

    %input variables definition
    Uanew{i}.name=’Ua’;
    Uanew{i}.type=’parameter’;
    Uanew{i}.val=Uamod{i};
    Uanew{i}.dim=3;
    Gbarnew{i}=Gbarnew{i};
    % Jnew{i} , Wnew{i} , Snew{i} , beta1new{i} , beta2new{i} , Gbarnew{i} , Wsubenew{i} , Wmasnew{i} , JSnew{i} , Ksnew{i} definitions
    ...
    %create matdataX.gdx file
    wgdx(['matdata' num2str(archivo) ],Jnew{i},Wnew{i},Snew{i},Uanew{i},
        beta1new{i}, beta2new{i}, Gbarnew{i}, Wsubenew{i},
        Wmasnew{i}, JSnew{i}, Ksnew{i});
    ...
    %execute gams
    gams(['logitCasoEstudio' num2str(archivo)]);
    %define unload structure (this part might be before the parfor)
    myVal{i}.name=’G’;
    myVal{i}.form=’sparse’;
    myVal2{i}.name=’Z’;
    myVal2{i}.form=’full’;
    %unload the variables
    G1{i}=rgdx(['matsol' num2str(archivo) ],myVal{i});
    Zvalue{i}=rgdx(['matsol' num2str(archivo) ],myVal2{i});
end
An optional point could be to delete the temporary files at the end of the execution:

```
%delete temporary files
delete('matdata.gdx');
delete('matdata.gms');
for i=1:particles
    delete(['matdata' num2str(i) '.gdx']);
    delete(['logitCasoEstudio' num2str(i) '.gms']);
    delete(['matsol' num2str(i) '.gdx']);
    delete(['logitCasoEstudio' num2str(i) '.lst']);
end
```

In some cases another problem encountered in some computers is that it could be necessary to execute the GMS temporary files one time without parallelizing them before all the parallel executions are made. At this moment the cause for this problem is unknown.